Quantum computing for nuclear physics

Part 1: Introduction to Quantum Computation

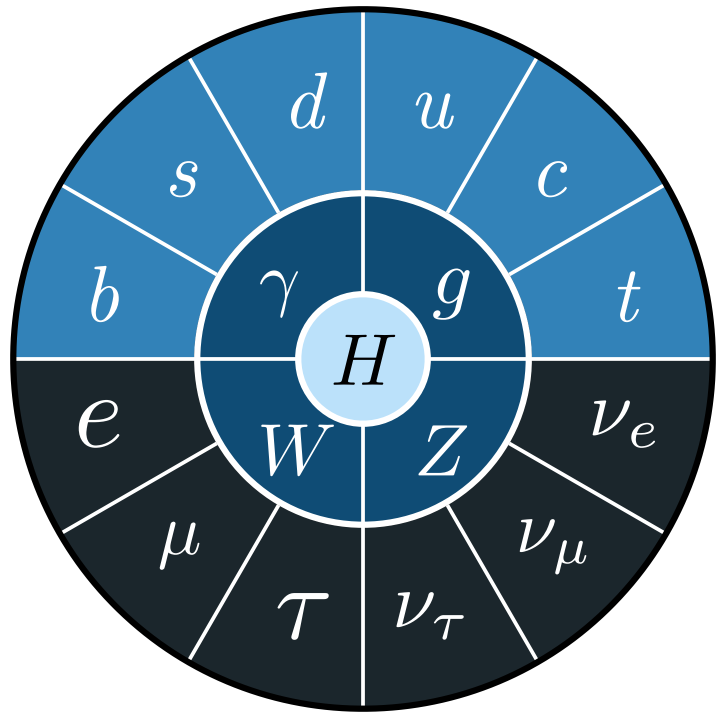
REYES Nuclear Mentoring Program
Aug 9, 2022

James Mulligan
University of California, Berkeley

Also see talk by D. Grabowska

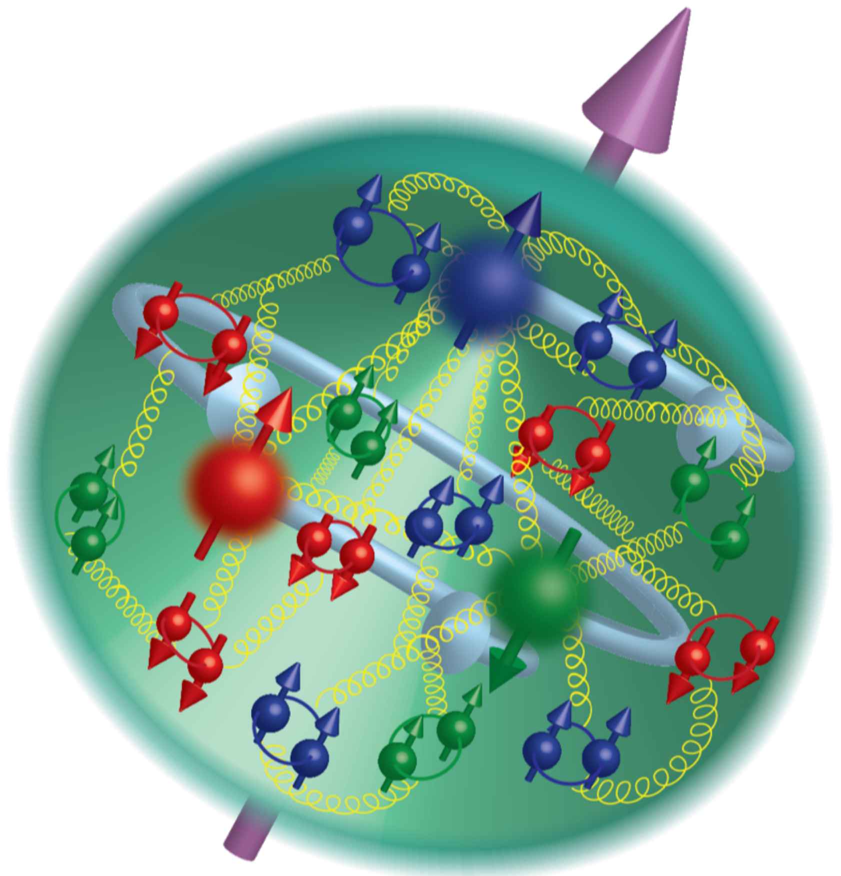
Quantum chromodynamics

The strong nuclear force — interactions of quarks and gluons



$$\mathscr{L} = -\frac{1}{4} F^a_{\mu\nu} F^{a\mu\nu} + \sum_{j=1}^{6} \bar{q}_j \left(i\gamma^\mu D_\mu - m_j \right) q_j$$

Coupling constant α_s



QCD interactions comprise $\approx 99\,\%$
of the visible mass in the universe!

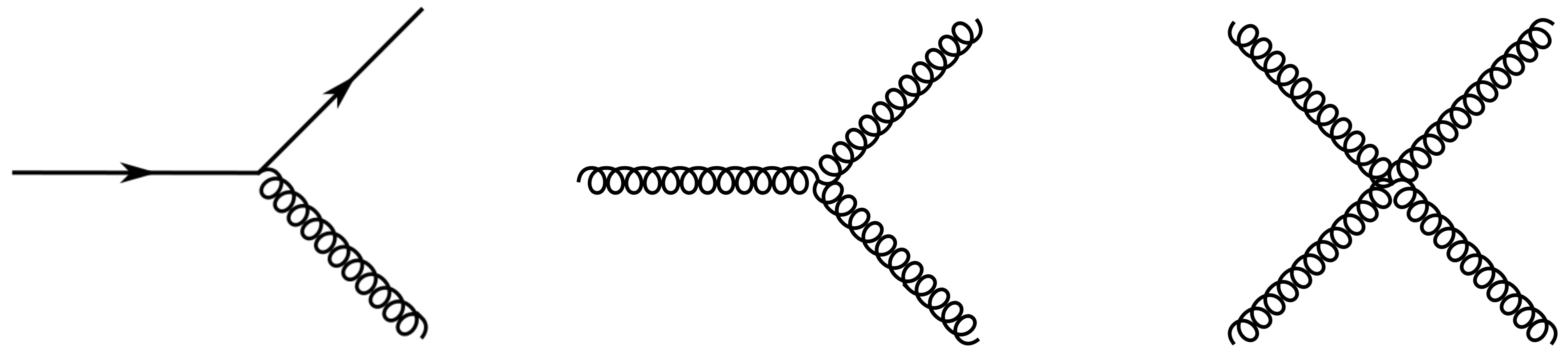
Can we solve the “equations of motion” given that we know the Lagrangian?

Solving the equations of motion

$$\mathscr{L} = -\frac{1}{4}F^a_{\mu\nu}F^{a\mu\nu} + \sum_{j=1}^{6}\bar{q}_j\left(i\gamma^\mu D_\mu - m_j\right)q_j$$

Perturbative QCD

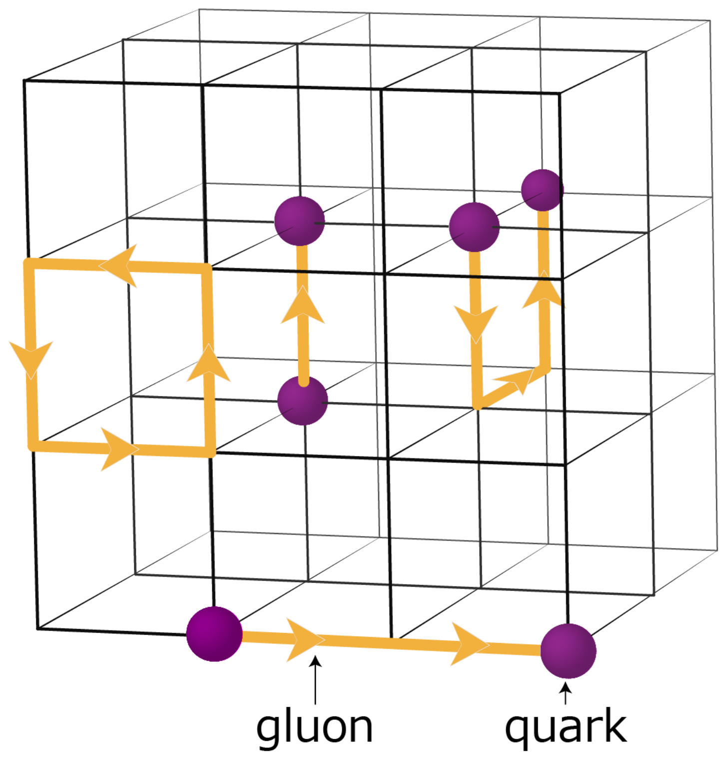
For $\alpha_s \ll 1$, compute scattering amplitudes with Feynman diagrams



$$\sigma = \sigma^{(0)} + \alpha_s\sigma^{(1)} + \alpha_s^2\sigma^{(2)} + \ldots$$

Lattice QCD

For low-density systems, compute static quantities with lattice regularization



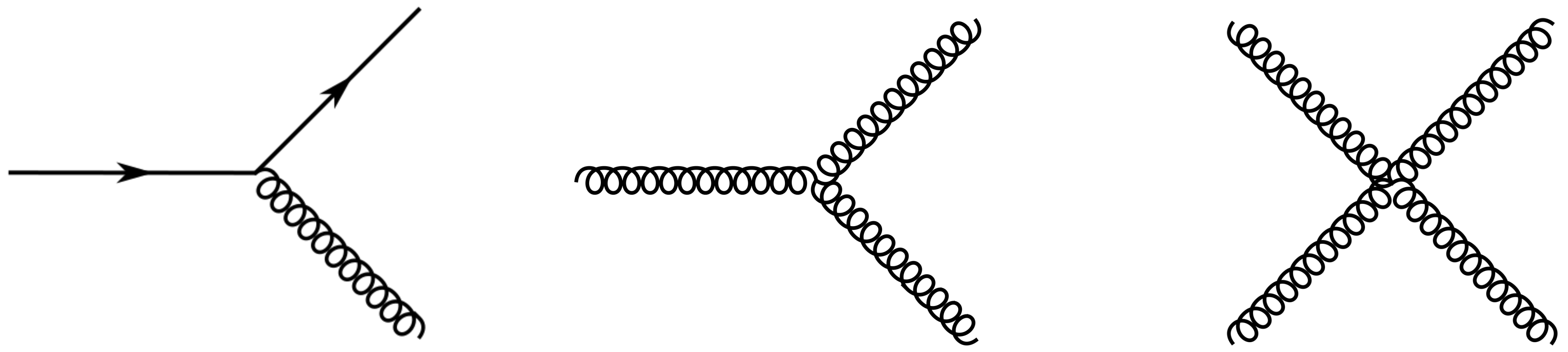
- Hadron spectra
- Deconfinement transition
- Chiral symmetry restoration

Solving the equations of motion

$$\mathscr{L} = -\frac{1}{4}F^a_{\mu\nu}F^{a\mu\nu} + \sum_{j=1}^{6} \bar{q}_j\left(i\gamma^\mu D_\mu - m_j\right)q_j$$

Perturbative QCD

For $\alpha_s \ll 1$, compute scattering amplitudes with Feynman diagrams

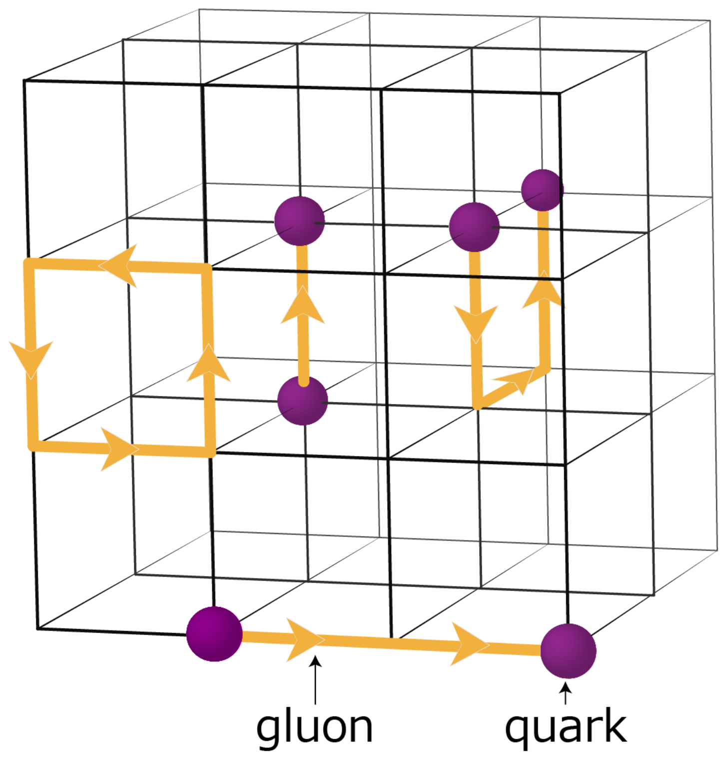


$$\sigma = \sigma^{(0)} + \alpha_s\sigma^{(1)} + \alpha_s^2\sigma^{(2)} + \ldots$$

...but no strong coupling!

Lattice QCD

For low-density systems, compute static quantities with lattice regularization



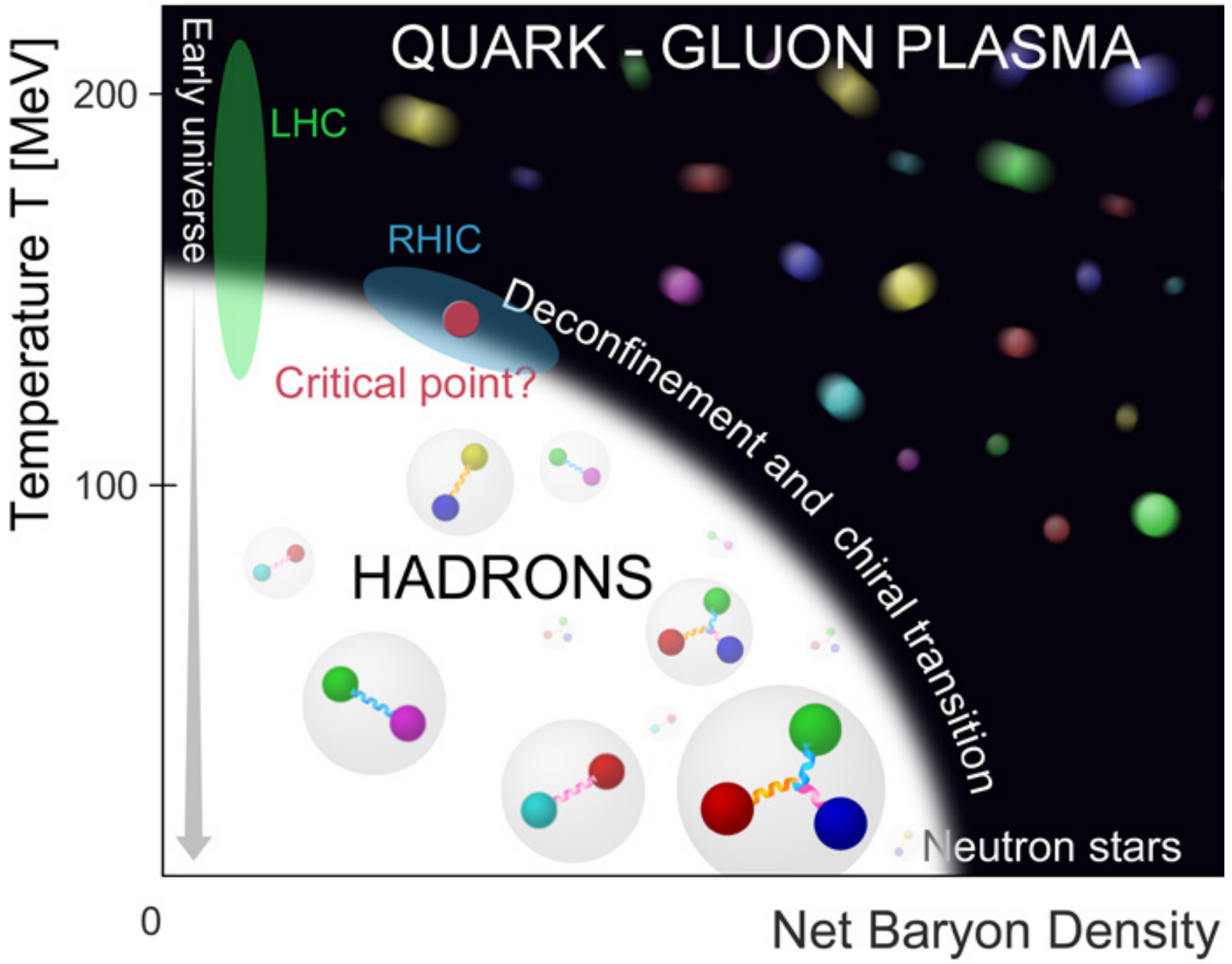
- Hadron spectra
- Deconfinement transition
- Chiral symmetry restoration

...but no dynamics!

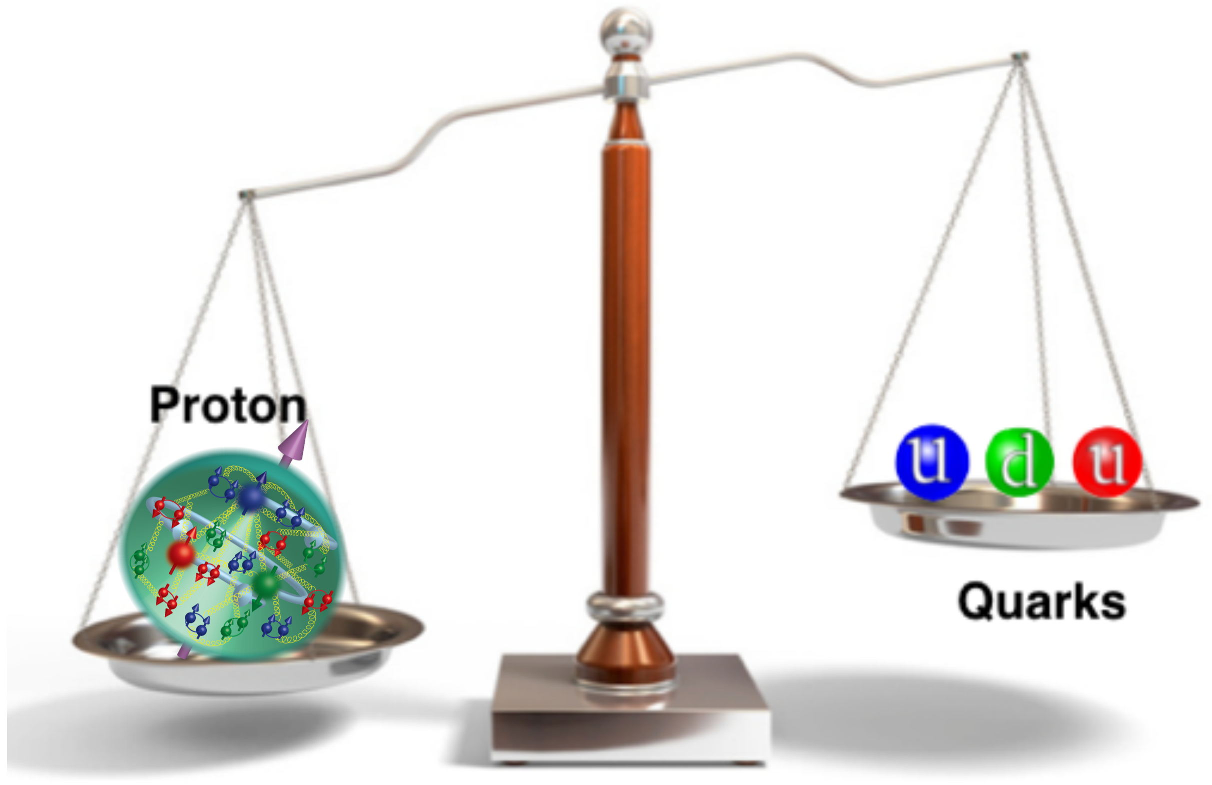
Emergence in QCD

While we know the Lagrangian…we still often don't know how to solve it

What is the landscape of QCD matter?



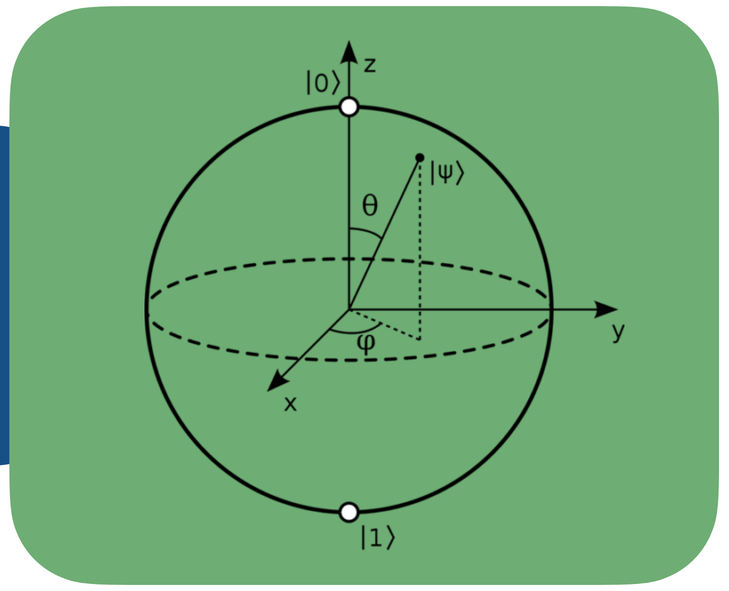
What are the *dynamics* that confine quarks and gluons into hadrons?



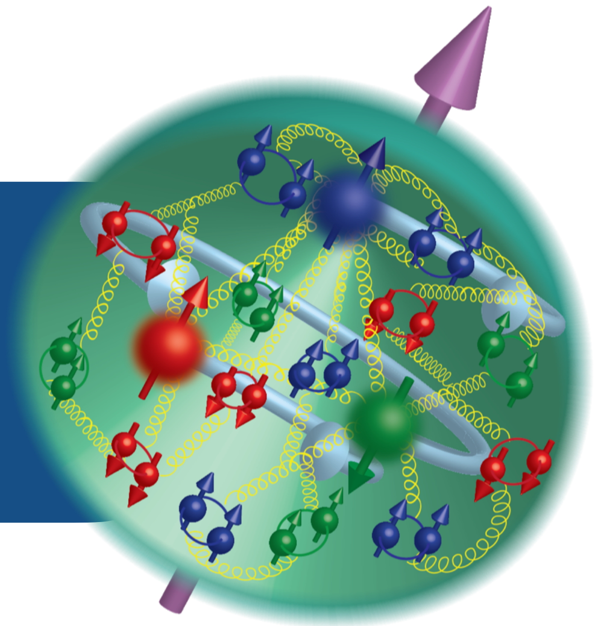
Opportunity to study a strongly-interacting, many-body quantum field theory

Outline

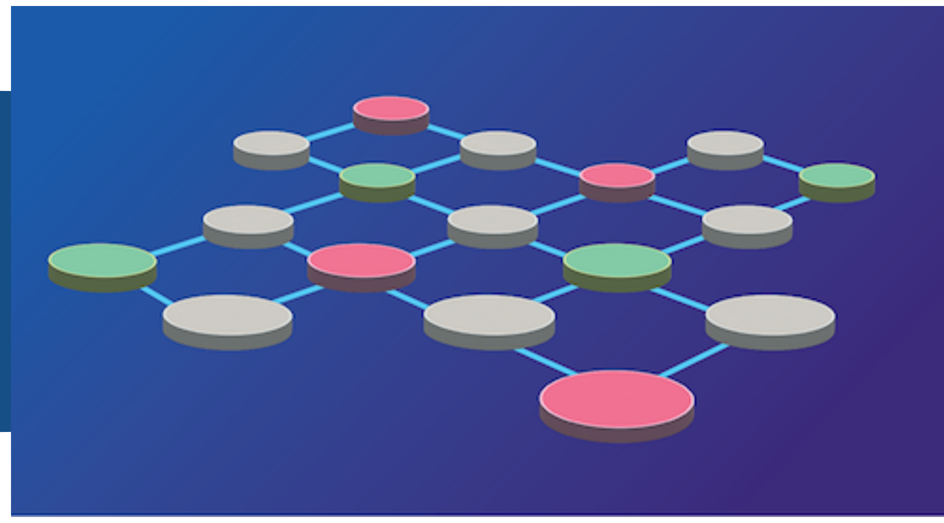
1. What is a quantum computer?



2. What are potential uses of quantum computing?

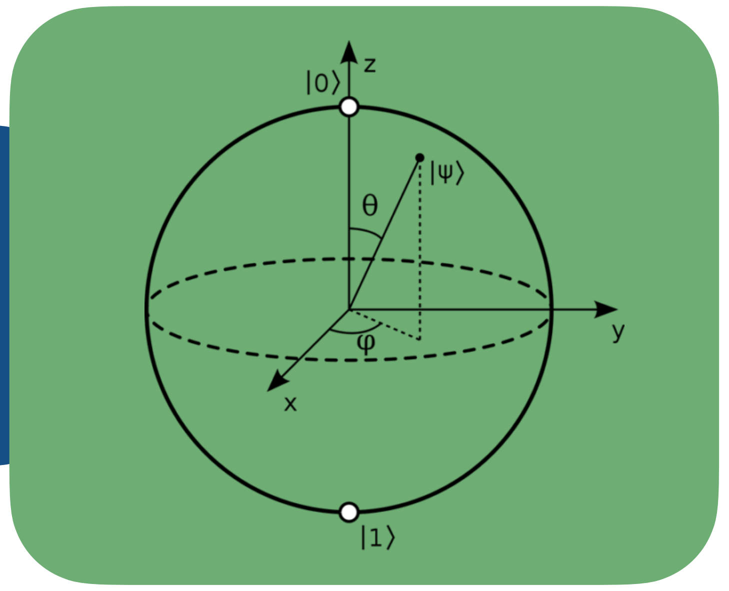


3. What are the challenges to achieve quantum advantage?



Outline

1. What is a quantum computer?



2. What are potential uses of quantum computing?

3. What are the challenges to achieve quantum advantage?

Computability

Given a scientific problem, there are two main questions we want to know:

Given infinite time and memory, can a computer eventually solve the problem?

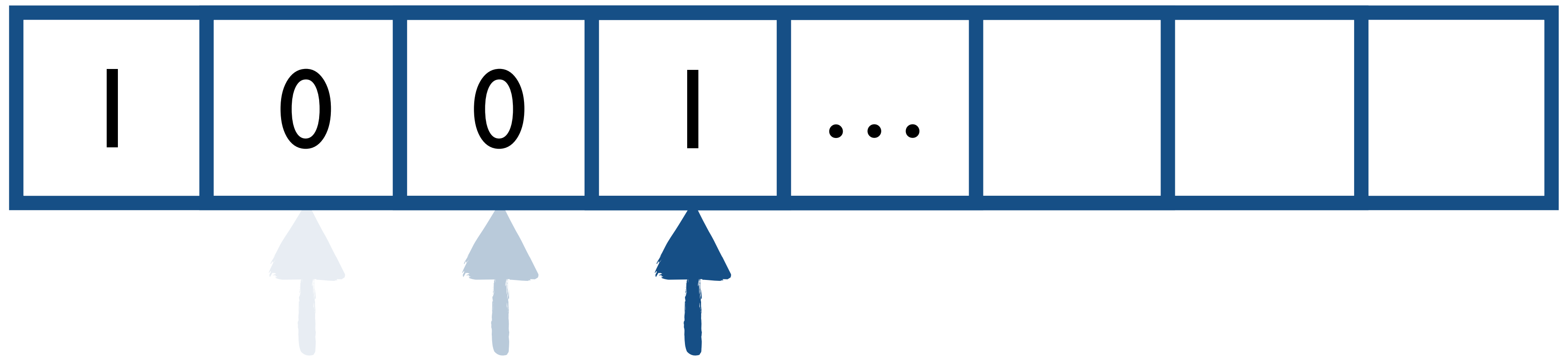
Can a computer find an answer in a reasonable amount of time?

Computability

Given a scientific problem, there are two main questions we want to know:

Given infinite time and memory, can a computer eventually solve the problem?

Example: The Halting Problem



Given an arbitrary computer program, is there an algorithm that can tell us whether the program will finish running? *No!*

Given infinite time, classical computers can solve the same problems as quantum computers

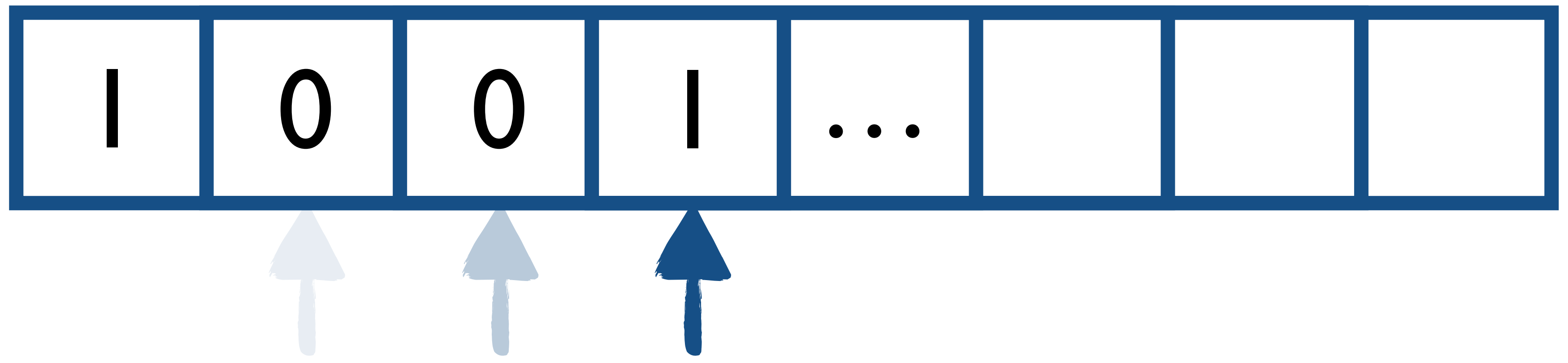
Can a computer find an answer in a reasonable amount of time?

Computability

Given a scientific problem, there are two main questions we want to know:

Given infinite time and memory, can a computer eventually solve the problem?

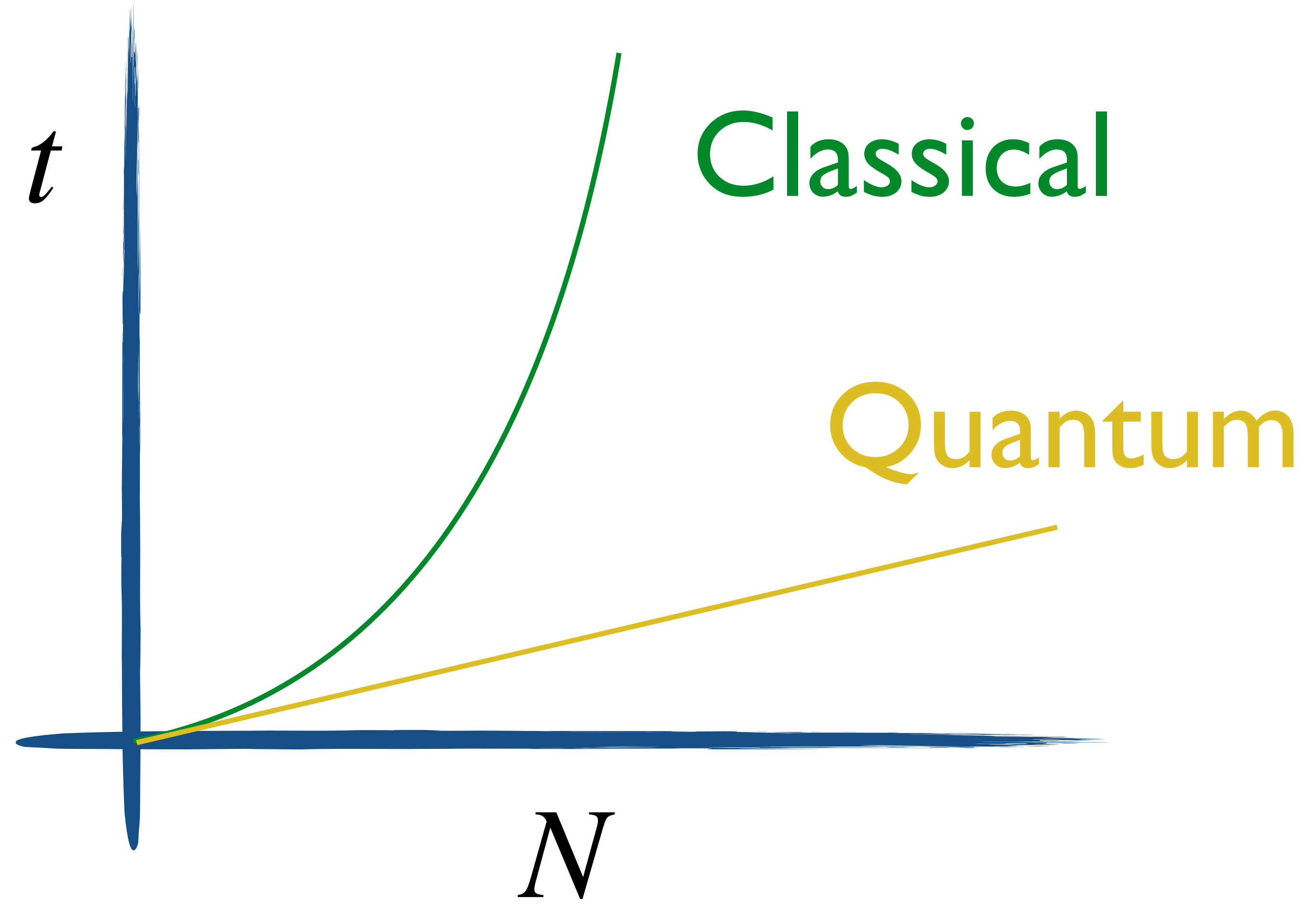
Example: The Halting Problem



Given an arbitrary computer program, is there an algorithm that can tell us whether the program will finish running? *No!*

Given infinite time, classical computers can solve the same problems as quantum computers

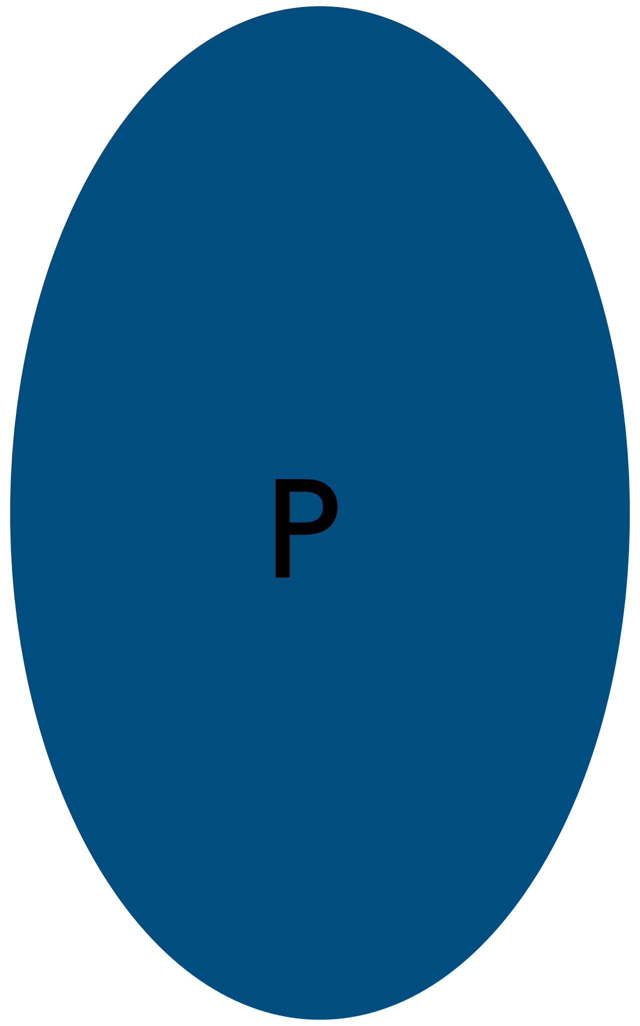
Can a computer find an answer in a reasonable amount of time?



But quantum computers can solve more problems *efficiently*

Computational complexity

P: Polynomial-time solution on classical computer



Example: matrix multiplication

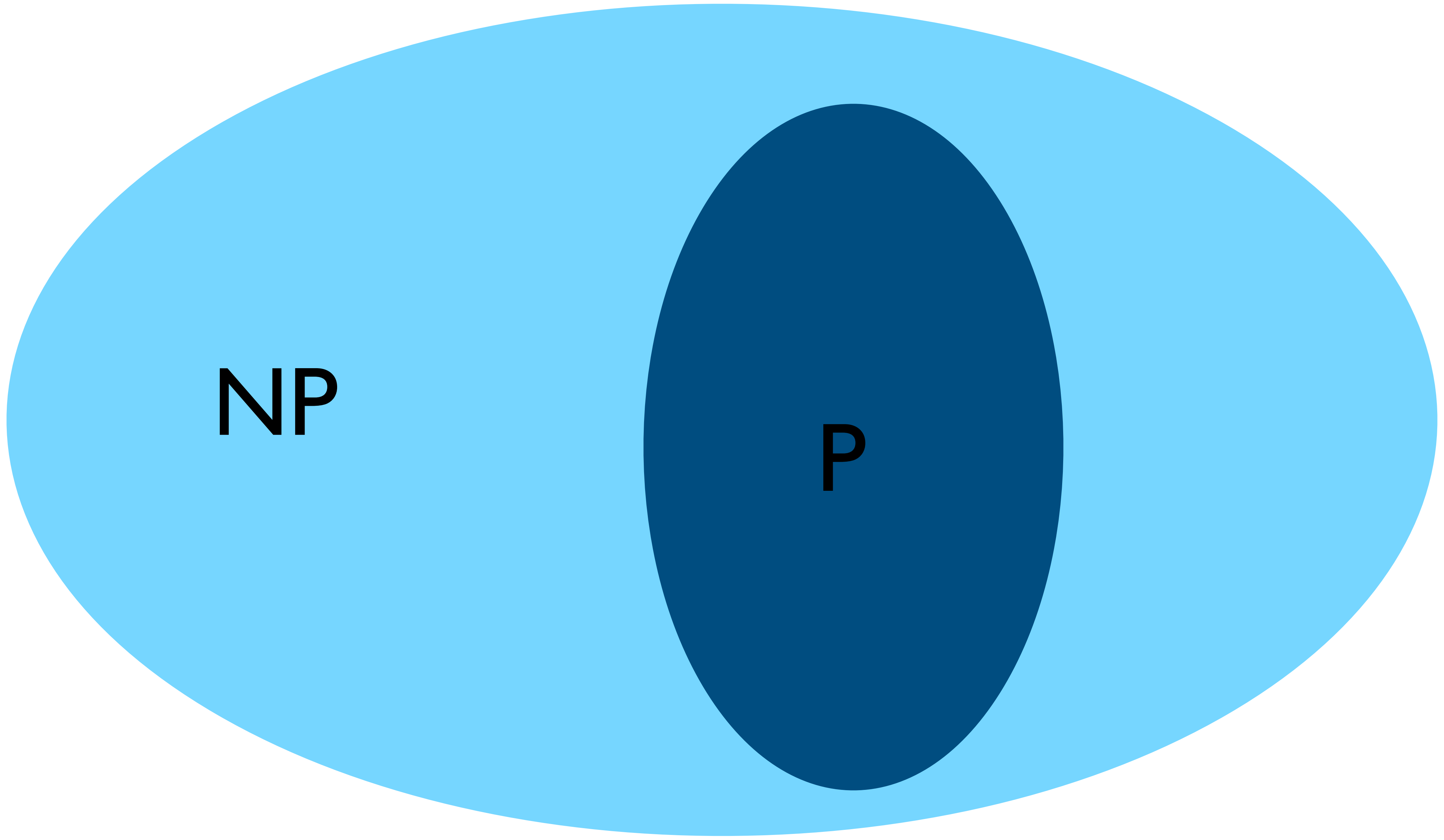
$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{np} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mp} \end{bmatrix}$$

Classical computer: $\text{Time} = \mathcal{O}\left(N^{2.373}\right)$

Computational complexity

P: Polynomial-time solution on classical computer

NP: Polynomial-time verification on classical computer

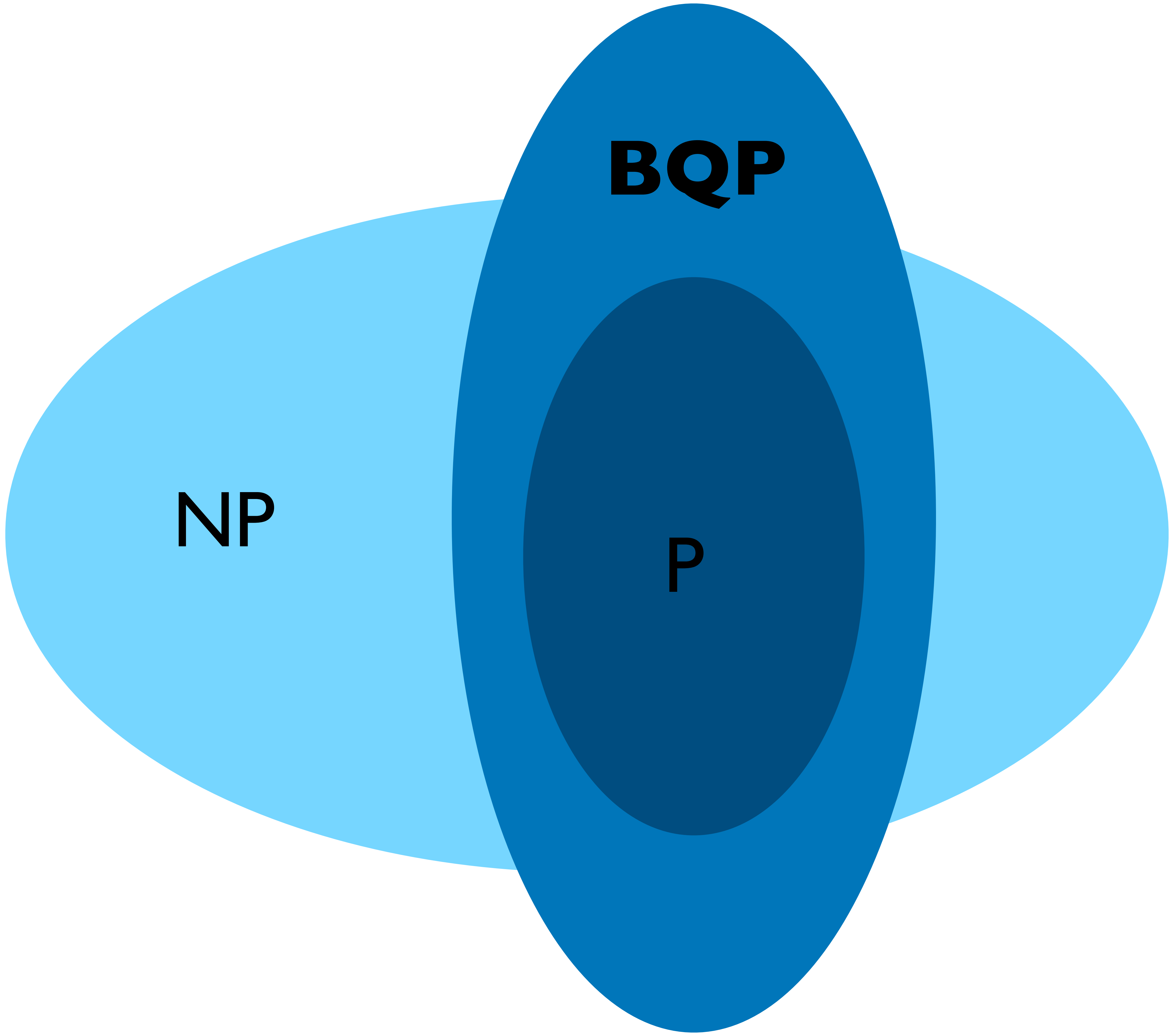


Example: Traveling salesman problem

What is the shortest route between N cities?

Classical computer: $\text{Time} = \mathcal{O}\left(\sim 2^N\right)$

Computational complexity

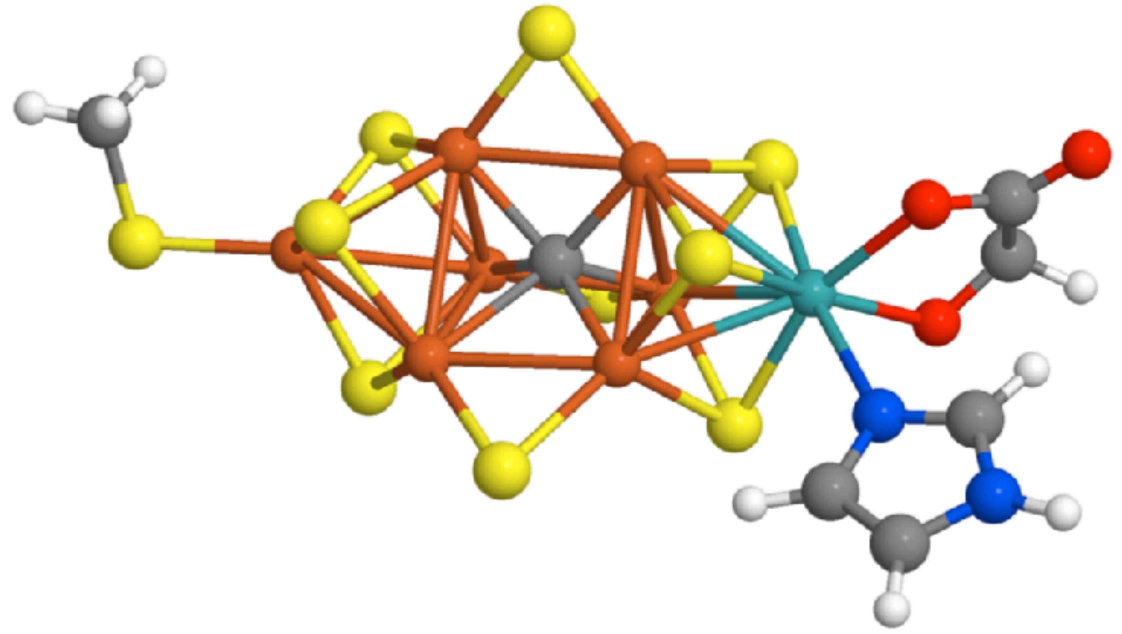


P: Polynomial-time solution on classical computer

NP: Polynomial-time verification on classical computer

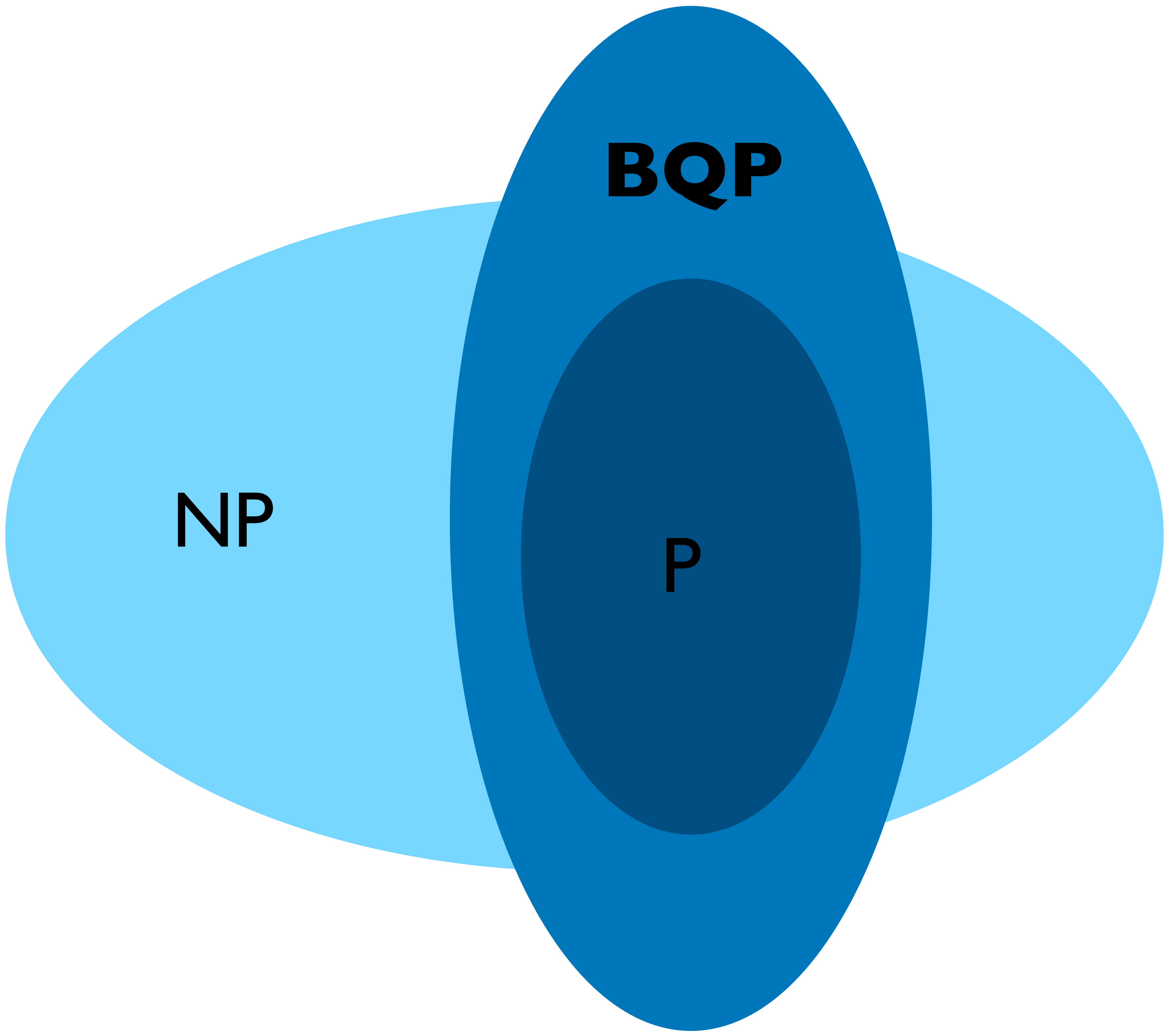
BQP: Polynomial-time solution on quantum computer

Example: Simulate the dynamics of an N-body quantum system



Classical computer: $\text{Time} = \mathcal{O}\left(\sim 2^N\right)$

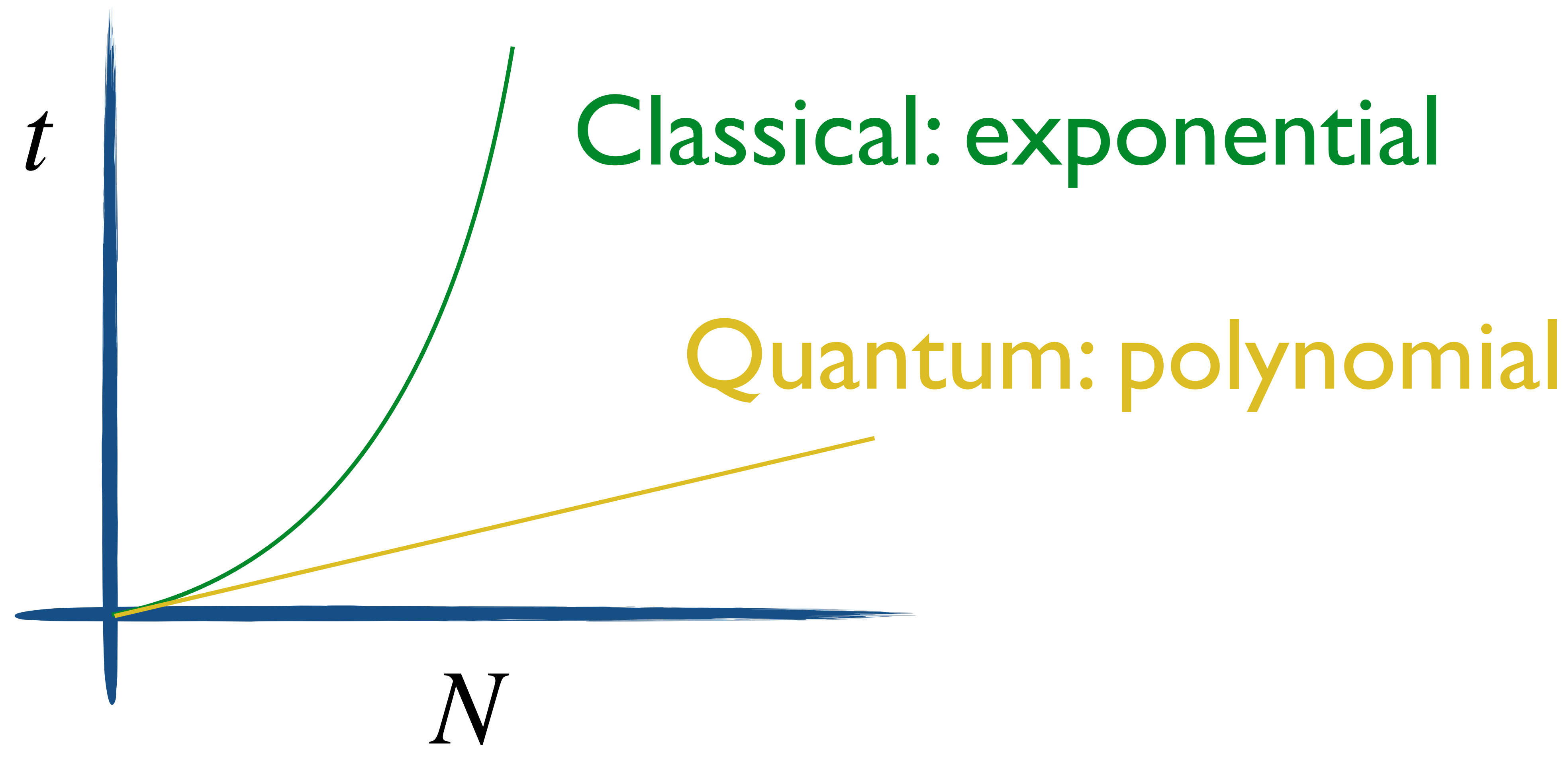
Computational complexity



P: Polynomial-time solution on classical computer

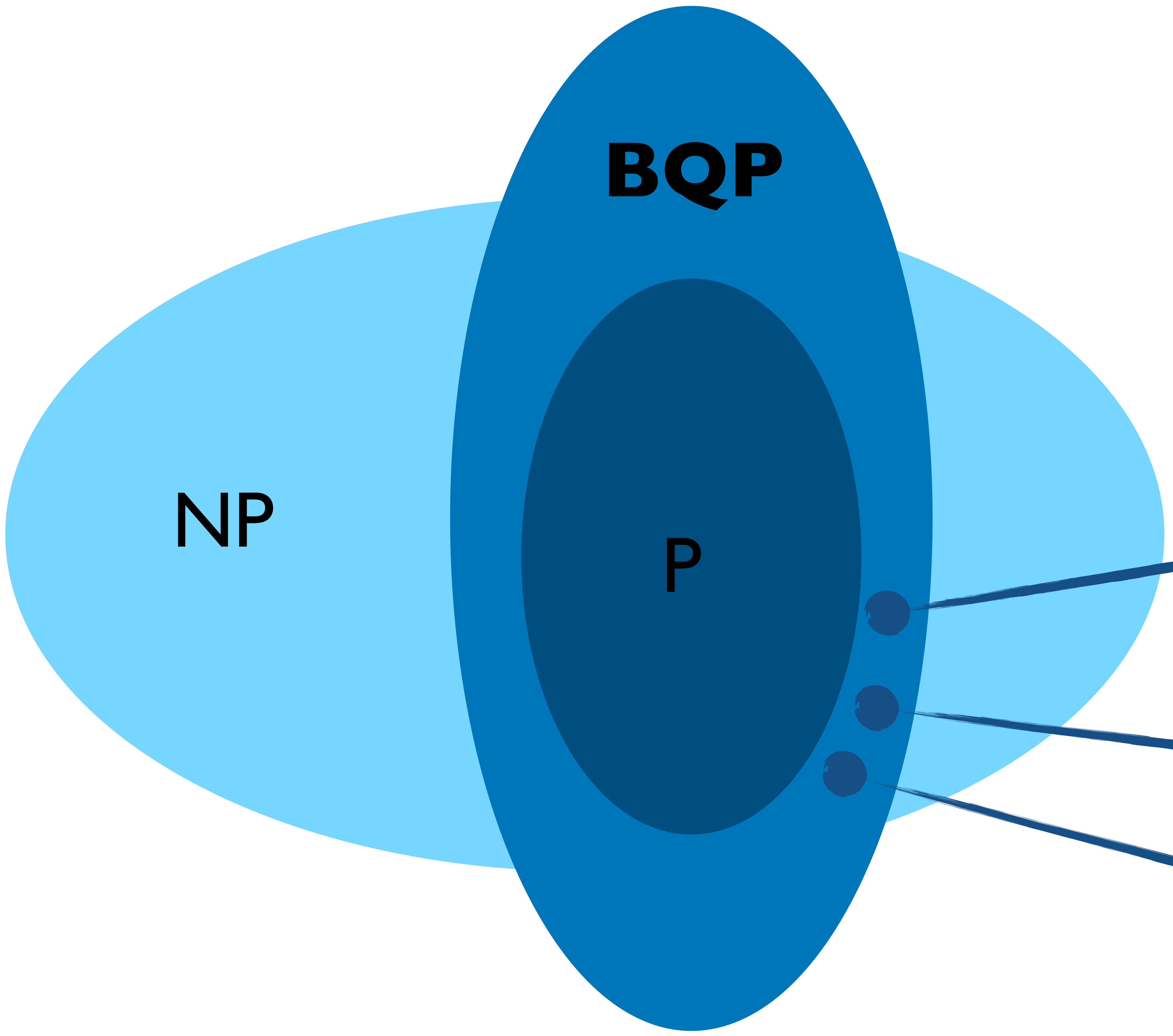
NP: Polynomial-time verification on classical computer

BQP: Polynomial-time solution on quantum computer



QC can solve *some* classically hard problems

Computational complexity



P: Polynomial-time solution on classical computer

NP: Polynomial-time verification on classical computer

BQP: Polynomial-time solution on quantum computer

Dynamics of many-body non-relativistic quantum system

Feynman (1982), Lloyd (1996)

Scattering in scalar field theory

Jordan, Lee, Preskill (2012-2018)

QCD?

Preskill (2018), Klco, Savage (2018), Muschik et al. (2016), Davoudi et al. (2019), …

Quantum advantage

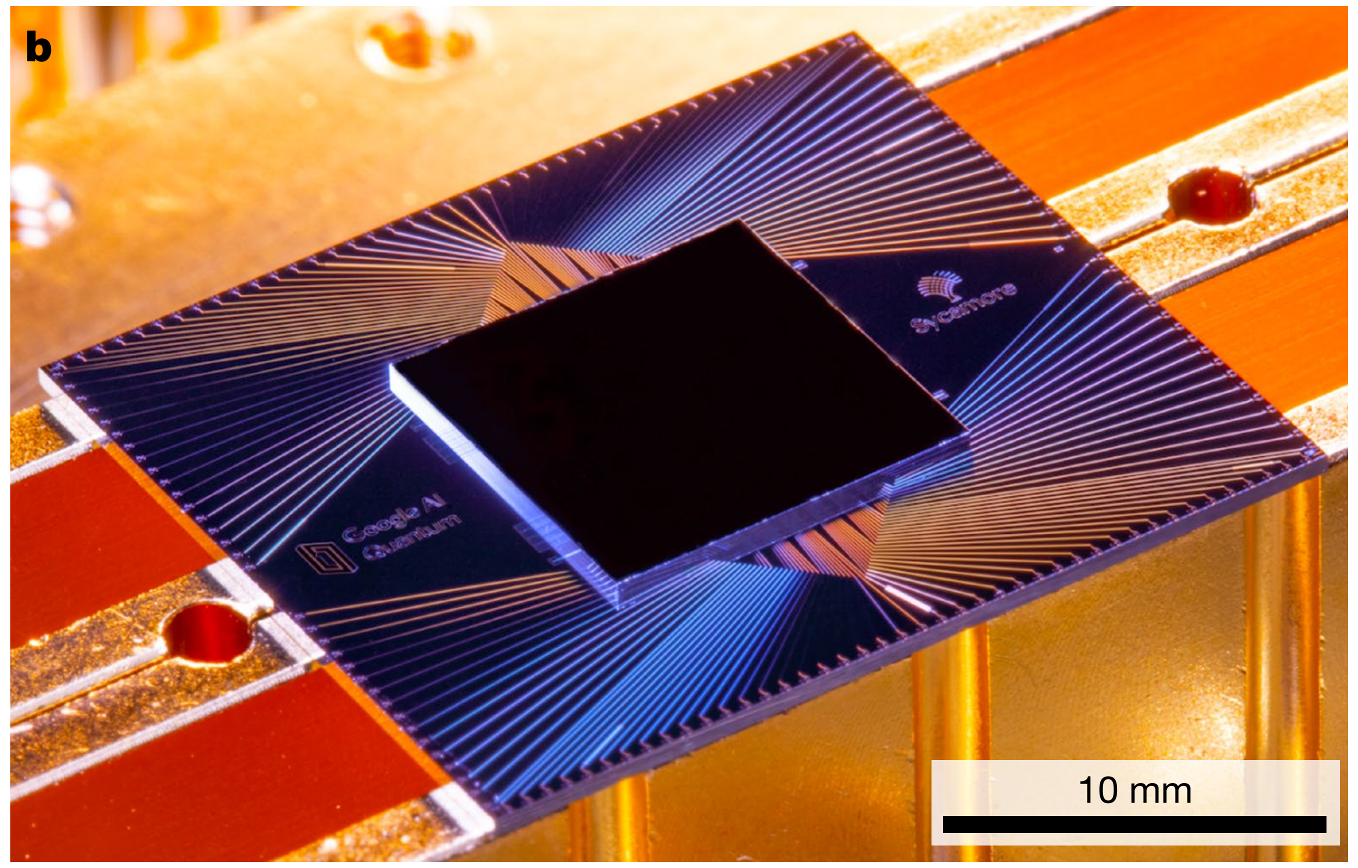
2019

Article

Quantum supremacy using a programmable superconducting processor

Google

Martinis et al., Nature (2019)



53-qubit superconducting circuit device

Algorithm: sampling of random circuits

$\mathcal{O}\left(10^3\right)$ times faster than best classical supercomputers

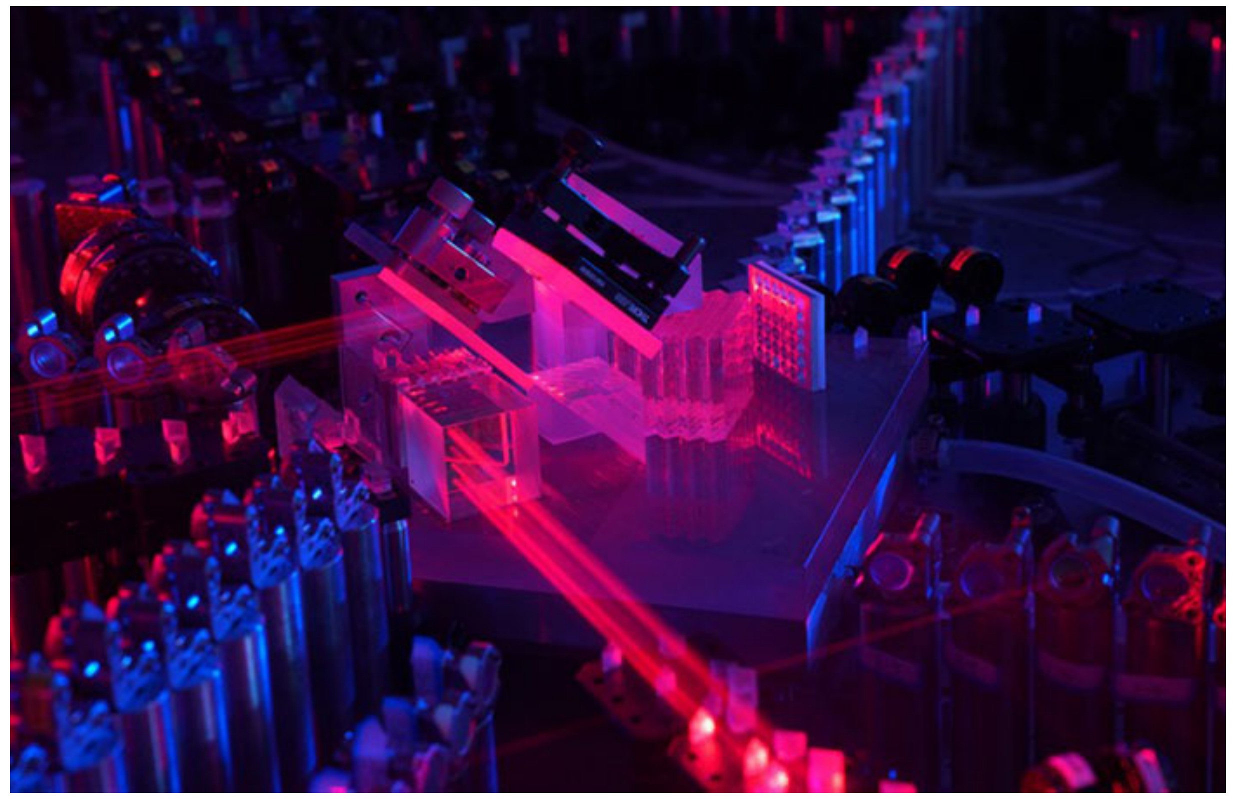
See also: Pan et al., PRL (2021)

2020-2021

Quantum computational advantage using photons

Han-Sen Zhong[1,2*], Hui Wang[1,2*], Yu-Hao Deng[1,2*], Ming-Cheng Chen[1,2*], Li-Chao Peng[1,2], Yi-Han Luo[1,2], Jian Qin[1,2], Dian Wu[1,2], Xing Ding[1,2], Yi Hu[1,2], Peng Hu[3], Xiao-Yan Yang[3], Wei-Jun Zhang[3], Hao Li[3], Yuxuan Li[4], Xiao Jiang[1,2], Lin Gan[4], Guangwen Yang[4], Lixing You[3], Zhen Wang[3], Li Li[1,2], Nai-Le Liu[1,2], Chao-Yang Lu[1,2], Jian-Wei Pan[1,2†]

Pan et al., Science (2020)



Photonic device — special-purpose

Algorithm: boson sampling

Claim: $\mathcal{O}\left(10^{14}\right)$ times faster than best classical supercomputers

Quantum advantage

Article

Qu… using a programmable su… essor

… 019)

Not so fast!

Ordinary computers can beat Google's quantum computer after all

Superfast algorithm put crimp in 2019 claim that Google's machine had achieved "quantum supremacy"

Aug 2022

https://www.science.org/content/article/ordinary-computers-can-beat-google-s-quantum-computer-after-all

10 mm

53-qubit superconducting circuit device

Algorithm: sampling of random circuits

$\mathcal{O}\left(10^3\right)$ times faster than best classical supercomputers

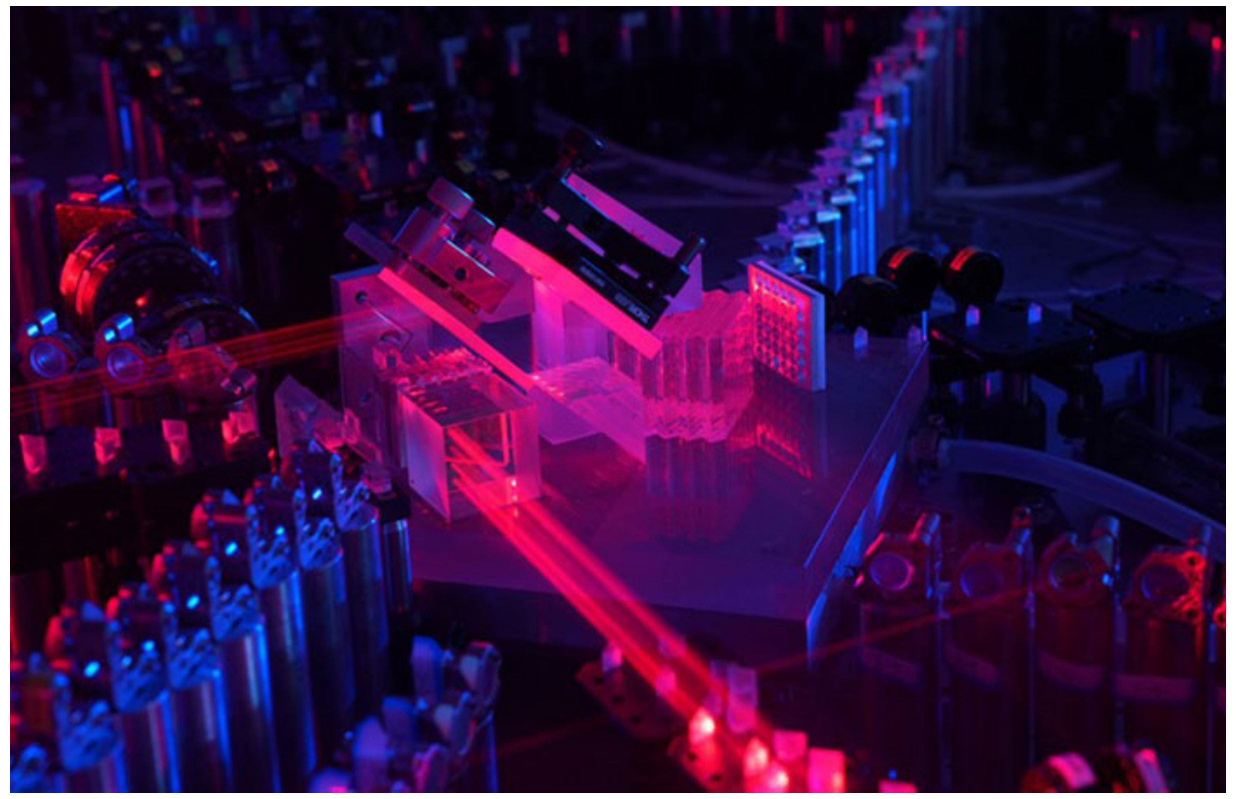
See also: Pan et al., PRL (2021)

2020-2021

Quantum computational advantage using photons

Han-Sen Zhong[1,2*], Hui Wang[1,2*], Yu-Hao Deng[1,2*], Ming-Cheng Chen[1,2*], Li-Chao Peng[1,2], Yi-Han Luo[1,2], Jian Qin[1,2], Dian Wu[1,2], Xing Ding[1,2], Yi Hu[1,2], Peng Hu[3], Xiao-Yan Yang[3], Wei-Jun Zhang[3], Hao Li[3], Yuxuan Li[4], Xiao Jiang[1,2], Lin Gan[4], Guangwen Yang[4], Lixing You[3], Zhen Wang[3], Li Li[1,2], Nai-Le Liu[1,2], Chao-Yang Lu[1,2], Jian-Wei Pan[1,2†]

Pan et al., Science (2020)



Photonic device — special-purpose

Algorithm: boson sampling

Claim: $\mathcal{O}\left(10^{14}\right)$ times faster than best classical supercomputers

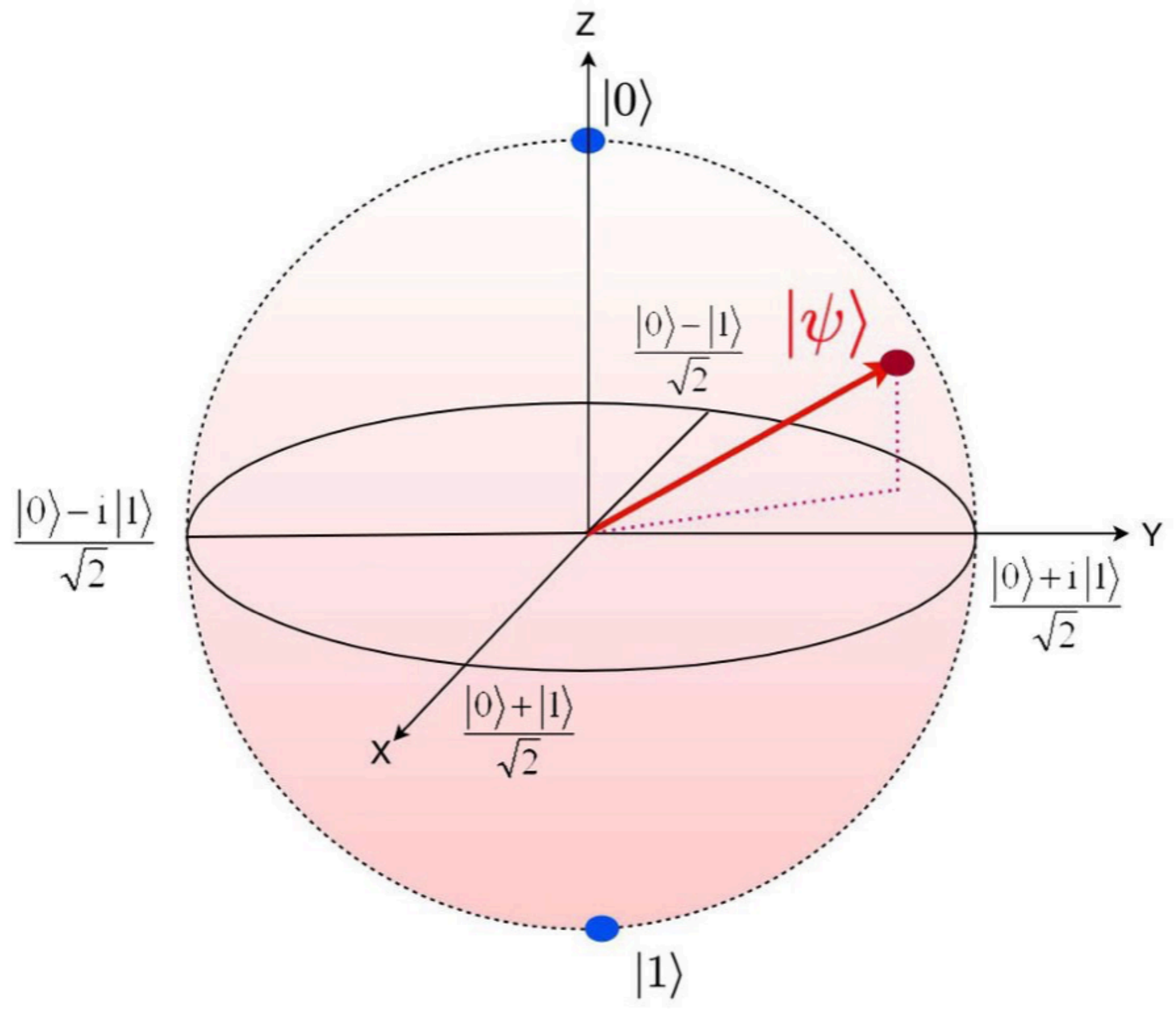
The building block of computation

Classical bit: 0 or 1

A classical bit can only be in one of two states

Quantum bit (qubit): $|\psi\rangle = a_0|0\rangle + a_1|1\rangle = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix}$

A quantum mechanical wave function that is a ***superposition*** of 0 and 1



When we measure the state $|\psi\rangle$, we obtain either:

- State $|0\rangle$, with a probability $|a_0|^2$
- State $|1\rangle$, with a probability $|a_1|^2$

A quantum bit can be in an infinite number of states, but when we measure it it can only be in one of two states

Multiple qubits

Quantum computations start to get interesting when we have *multiple* qubits

We represent a quantum state of multiple qubits using a ***tensor product***:

Two qubits:

$$|a\rangle = a_0|00\rangle + a_1|01\rangle + a_2|10\rangle + a_3|11\rangle = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

where $|00\rangle = |0\rangle \otimes |0\rangle$ denotes the tensor product

N qubits:

$$|a\rangle = a_0|0...0\rangle + \ldots + a_K|1...1\rangle = \begin{pmatrix} a_0 \\ \vdots \\ a_K \end{pmatrix}$$

Multiple qubits

Quantum computations start to get interesting when we have *multiple* qubits

We represent a quantum state of multiple qubits using a ***tensor product***:

Example:

$$|01\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \times 0 \\ 1 \times 1 \\ 0 \times 0 \\ 0 \times 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$|\psi_{ab}\rangle = |a\rangle \otimes |b\rangle = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} \otimes \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} = \begin{pmatrix} a_0 b_0 \\ a_0 b_1 \\ a_1 b_0 \\ a_1 b_1 \end{pmatrix}$$

Where does quantum advantage come from?

$$|\psi\rangle = \sum_{i=1}^{2^N} a_i |\psi_i\rangle$$

For N qubits, there are 2^N amplitudes

e.g. $|\psi\rangle = a_1|000\rangle + a_2|001\rangle + a_3|010\rangle + a_4|011\rangle + a_5|100\rangle + a_6|101\rangle + a_7|110\rangle + a_8|111\rangle$

Where does quantum advantage come from?

$$|\psi\rangle = \sum_{i=1}^{2^N} a_i |\psi_i\rangle$$

For N qubits, there are 2^N amplitudes

e.g. $|\psi\rangle = a_1|000\rangle + a_2|001\rangle + a_3|010\rangle + a_4|011\rangle + a_5|100\rangle + a_6|101\rangle + a_7|110\rangle + a_8|111\rangle$

A quantum operation (gate) modifies all of these 2^N amplitudes simultaneously!

$$|a\rangle = \sum_{i=1}^{2^N} a_i |\psi_i\rangle \rightarrow |b\rangle = \sum_{i=1}^{2^N} b_i |\psi_i\rangle$$

Where does quantum advantage come from?

$$|\psi\rangle = \sum_{i=1}^{2^N} a_i |\psi_i\rangle$$

For N qubits, there are 2^N amplitudes

e.g. $$|\psi\rangle = a_1|000\rangle + a_2|001\rangle + a_3|010\rangle + a_4|011\rangle + a_5|100\rangle + a_6|101\rangle + a_7|110\rangle + a_8|111\rangle$$

A quantum operation (gate) modifies all of these 2^N amplitudes simultaneously!

$$|a\rangle = \sum_{i=1}^{2^N} a_i |\psi_i\rangle \rightarrow |b\rangle = \sum_{i=1}^{2^N} b_i |\psi_i\rangle$$

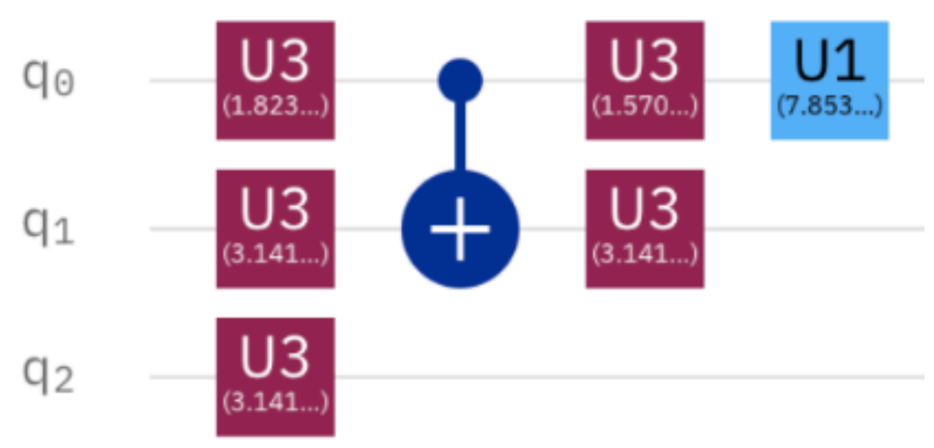
However: we cannot access the quantum amplitudes $\{a_i\}$ directly!

This is the major challenge: How can we take advantage of the exponential efficiency of quantum operations when we only access one randomly sampled state at a time?

Quantum computing

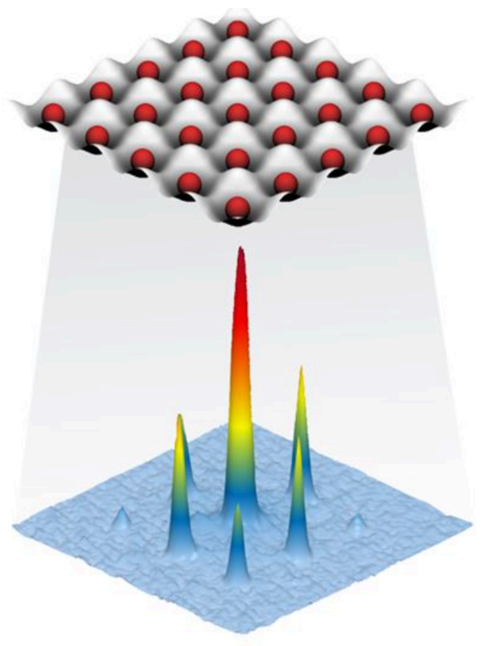
Digital quantum computers

Universal



Analog quantum computers

Application-specific



In this program we will focus on digital, universal quantum computers

→ Quantum circuits: qubits + quantum logic gates

Quantum Gates

Quantum states: column vectors $|a\rangle = a_0|0\rangle + a_1|1\rangle = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix}$

Quantum logic gates: unitary matrices $U^\dagger U = I$

- Single-qubit gates:

$$\alpha|0\rangle + \beta|1\rangle \xrightarrow{\;X\;} \beta|0\rangle + \alpha|1\rangle \qquad X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$\alpha|0\rangle + \beta|1\rangle \xrightarrow{\;Z\;} \alpha|0\rangle - \beta|1\rangle \qquad Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$\alpha|0\rangle + \beta|1\rangle \xrightarrow{\;H\;} \alpha\frac{|0\rangle+|1\rangle}{\sqrt{2}} + \beta\frac{|0\rangle-|1\rangle}{\sqrt{2}} \qquad H \equiv \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- Multi-qubit gates: Any quantum gate can be composed from CNOT and single-qubit gates

Example: CNOT Gate

Definition: if first qubit is 1, then flip the state of the second qubit

$$|00\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \rightarrow \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \qquad |01\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix} \rightarrow \begin{pmatrix}0\\1\\0\\0\end{pmatrix}$$

$$|10\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \rightarrow \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \qquad |11\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \rightarrow \begin{pmatrix}0\\0\\1\\0\end{pmatrix}$$

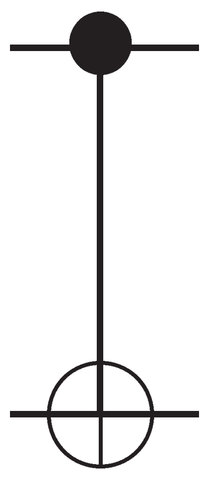
Example: CNOT Gate

Definition: if first qubit is 1, then flip the state of the second qubit

$$|00\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \rightarrow \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \qquad |01\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix} \rightarrow \begin{pmatrix}0\\1\\0\\0\end{pmatrix}$$

$$|10\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \rightarrow \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \qquad |11\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \rightarrow \begin{pmatrix}0\\0\\1\\0\end{pmatrix}$$

How do we construct the matrix representation of the gate?



$$= \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{bmatrix}$$

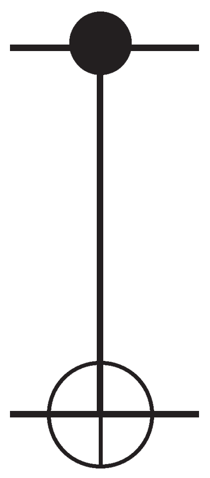
Example: CNOT Gate

Definition: if first qubit is 1, then flip the state of the second qubit

$$\boxed{|00\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \rightarrow \begin{pmatrix}1\\0\\0\\0\end{pmatrix}} \qquad |01\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix} \rightarrow \begin{pmatrix}0\\1\\0\\0\end{pmatrix}$$

$$|10\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \rightarrow \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \qquad |11\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \rightarrow \begin{pmatrix}0\\0\\1\\0\end{pmatrix}$$

How do we construct the matrix representation of the gate?



$$= \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{bmatrix}$$

The i^{th} column determines how the i^{th} basis state transforms:

$$|00\rangle \rightarrow |00\rangle$$

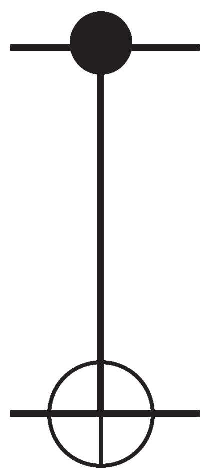
Example: CNOT Gate

Definition: if first qubit is 1, then flip the state of the second qubit

$$|00\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \rightarrow \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \qquad |01\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix} \rightarrow \begin{pmatrix}0\\1\\0\\0\end{pmatrix}$$

$$|10\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \rightarrow \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \qquad |11\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \rightarrow \begin{pmatrix}0\\0\\1\\0\end{pmatrix}$$

How do we construct the matrix representation of the gate?



$$= \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{bmatrix}$$

The i^{th} column determines how the i^{th} basis state transforms:

$$|01\rangle \rightarrow |01\rangle$$

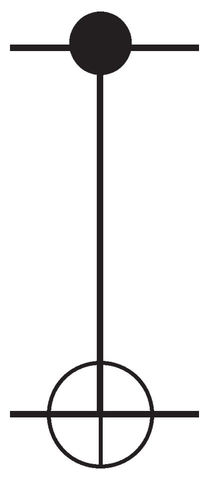
Example: CNOT Gate

Definition: if first qubit is 1, then flip the state of the second qubit

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$\boxed{|10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}} \qquad |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

How do we construct the matrix representation of the gate?



$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

The i^{th} column determines how the i^{th} basis state transforms:

$$|10\rangle \rightarrow |11\rangle$$

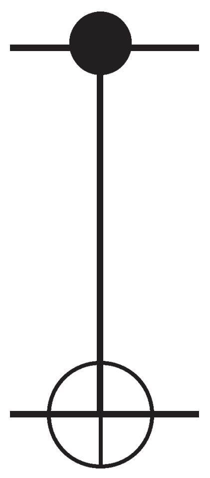
Example: CNOT Gate

Definition: if first qubit is 1, then flip the state of the second qubit

$$|00\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \to \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \qquad |01\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix} \to \begin{pmatrix}0\\1\\0\\0\end{pmatrix}$$

$$|10\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \to \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \qquad \boxed{|11\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \to \begin{pmatrix}0\\0\\1\\0\end{pmatrix}}$$

How do we construct the matrix representation of the gate?



$$= \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{bmatrix}$$

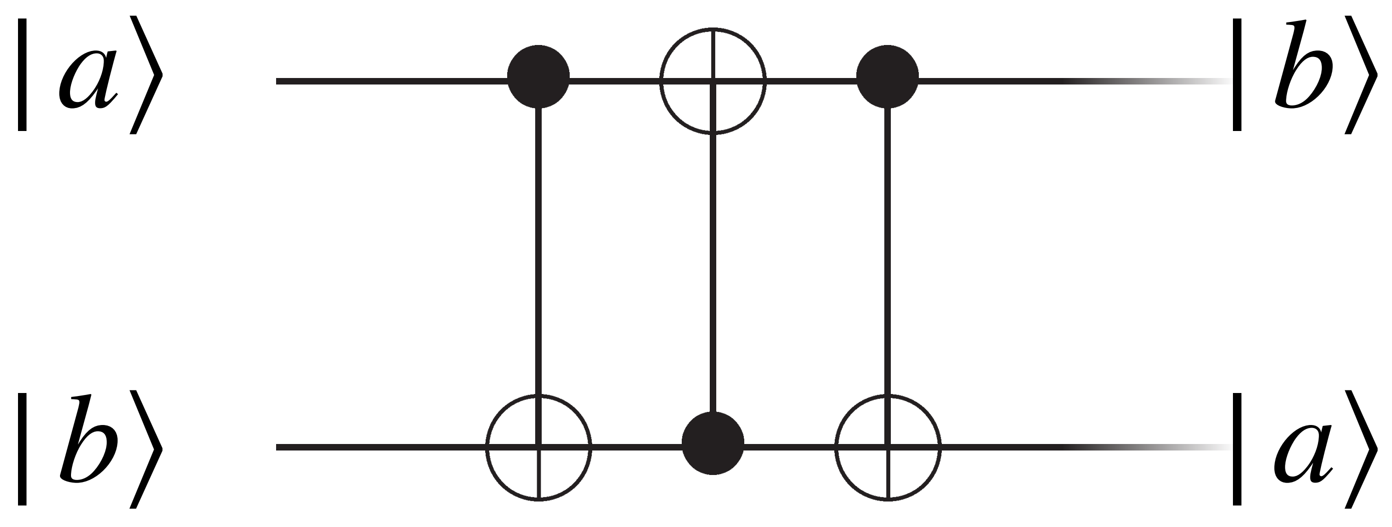
The i^{th} column determines how the i^{th} basis state transforms:

$$|11\rangle \to |10\rangle$$

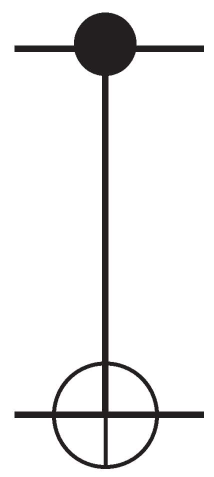
Quantum circuits

Nothing more than (clever) unitary matrix multiplications!

Example: SWAP circuit



where



$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

CNOT gate

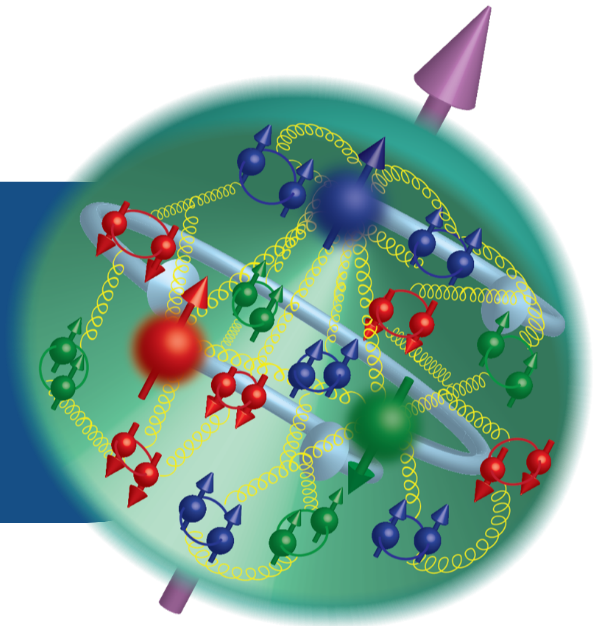
$$SWAP\left(|a\rangle \otimes |b\rangle\right) = CNOT_{0,1} \times CNOT_{1,0} \times CNOT_{0,1} \times \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} \otimes \begin{pmatrix} b_0 \\ b_1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} a_0 b_0 \\ a_0 b_1 \\ a_1 b_0 \\ a_1 b_1 \end{pmatrix} = \begin{pmatrix} b_0 a_0 \\ b_0 a_1 \\ b_1 a_0 \\ b_1 a_1 \end{pmatrix} = |b\rangle \otimes |a\rangle$$

Outline

1. What is a quantum computer?

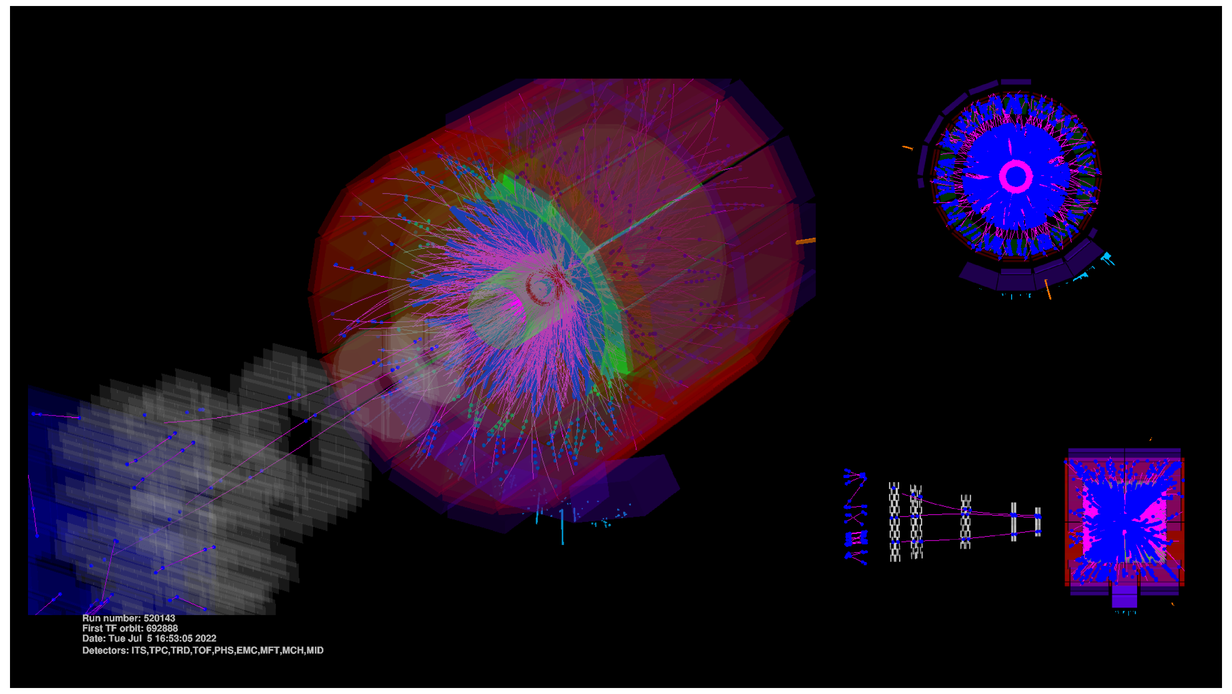
2. What are potential uses of quantum computing?

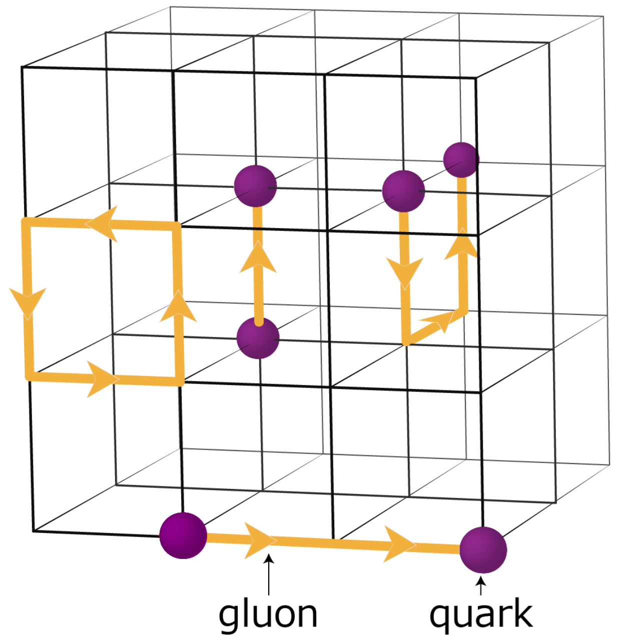


3. What are the challenges to achieve quantum advantage?

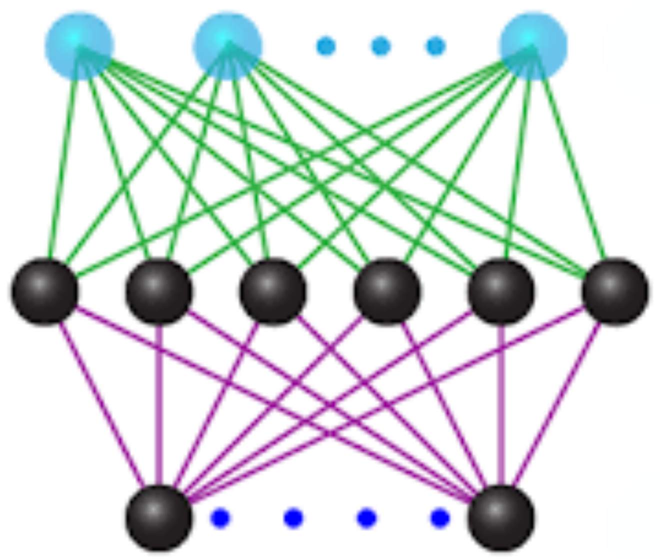
Future applications of quantum computers

Simulation of quantum field theory

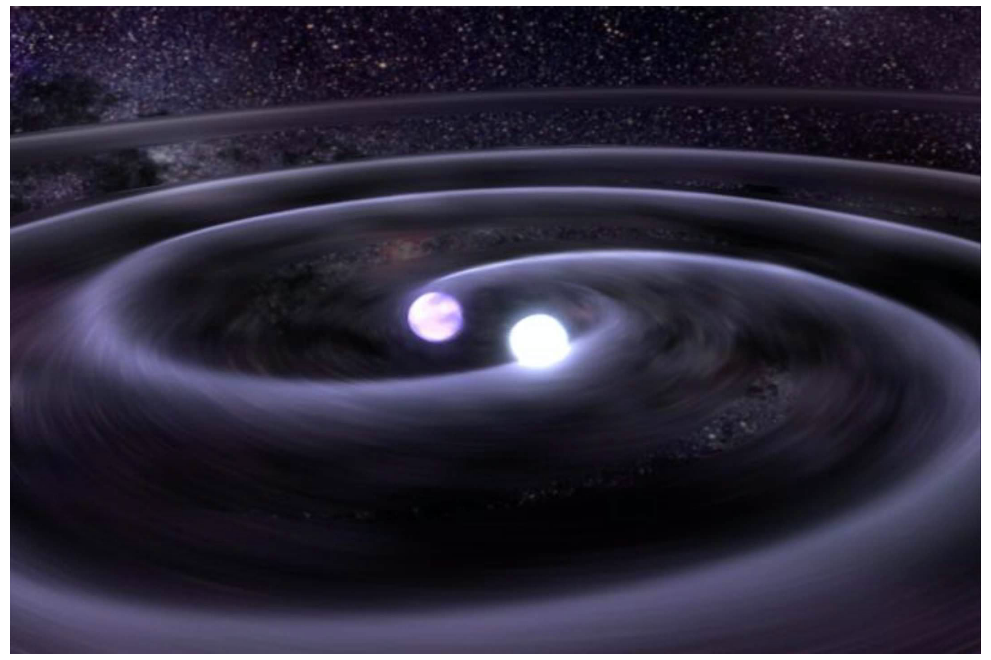




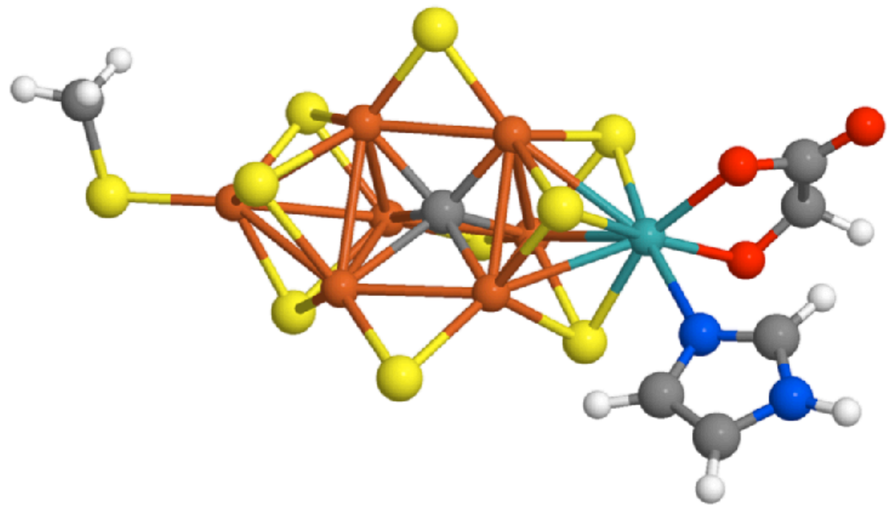
Quantum machine learning



Dense nuclear matter



Molecular dynamics



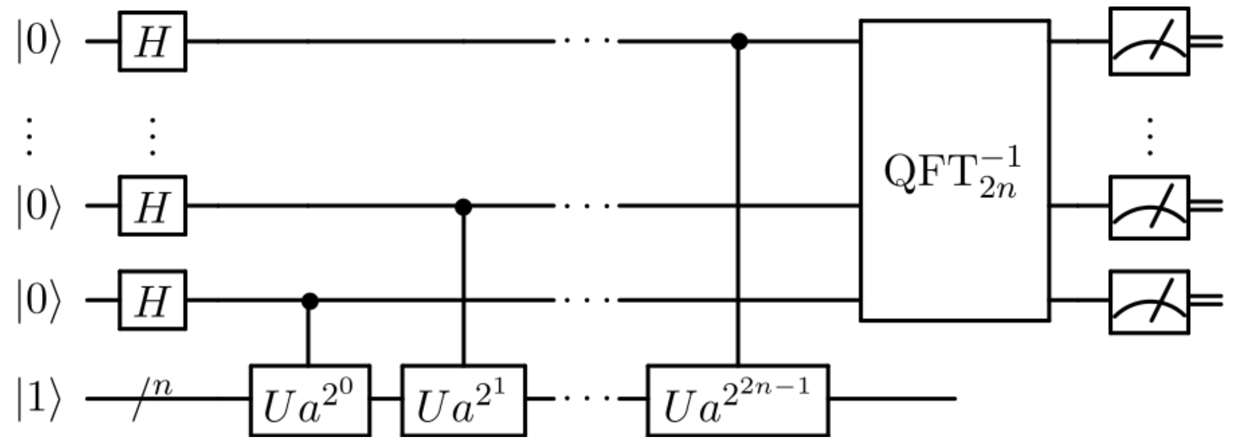
Cryptography

…

Quantum algorithms

Shor's factoring algorithm

Task: Find prime factors of an integer

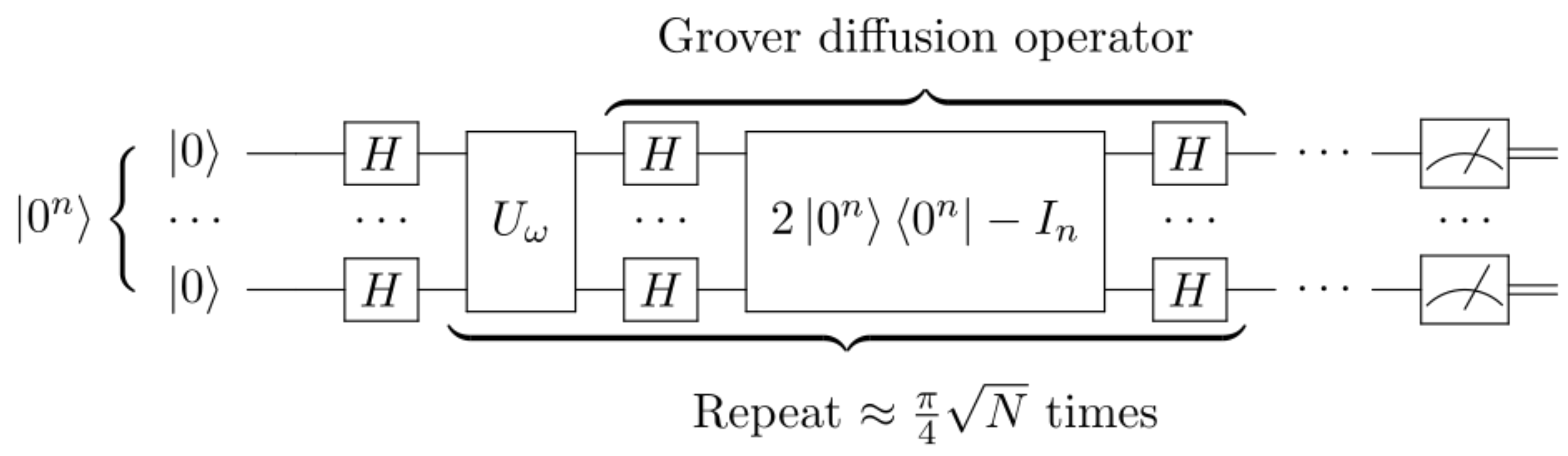


Exponential speedup compared to classical algorithms

$\mathcal{O}\left((\log N)^2 \ldots\right)$ vs. $\mathcal{O}\left(e^{1.9(\log N)^{1/3}\ldots}\right)$

Grover's search algorithm

Task: Find marked entry in an unordered list



Polynomial speedup compared to classical algorithms

$\mathcal{O}\left(\sqrt{(N)}\right)$ vs. $\mathcal{O}(N)$

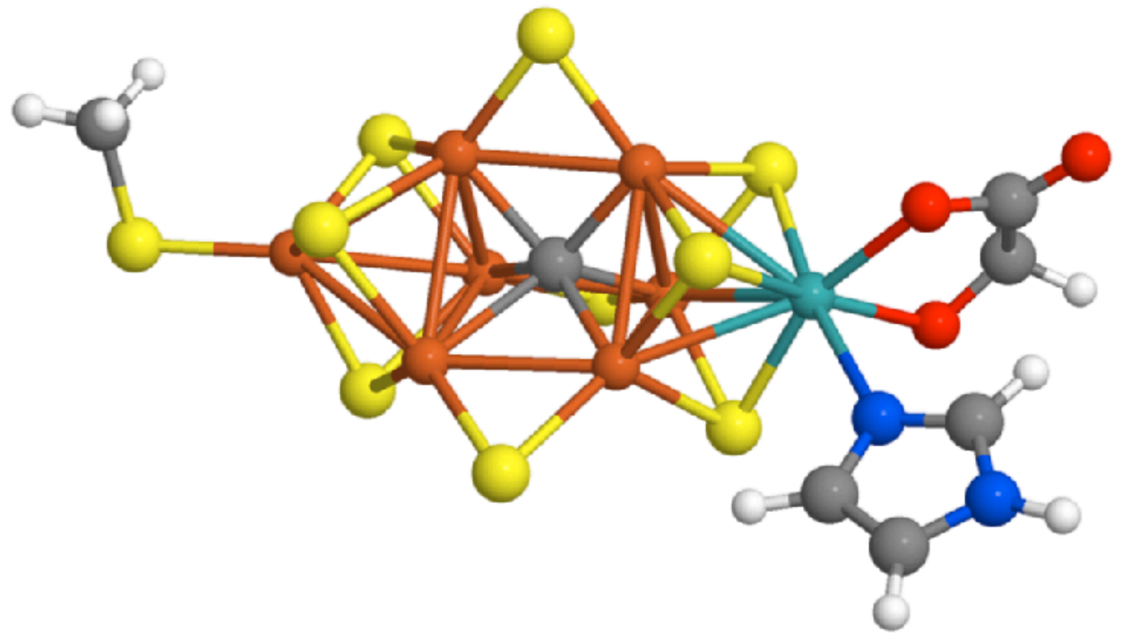
And more…

Quantum simulation

Feynman `81
Lloyd `96

Task: Given the Hamiltonian of a quantum mechanical system, simulate its dynamical evolution

- Quantum chemistry, material design, nuclear dynamics, …



That is, solve the time-dependent Schrödinger equation:

$$H|\psi(t)\rangle = i\hbar\frac{d}{dt}|\psi(t)\rangle$$

The solution is just a unitary evolution!

$$|\psi(t)\rangle = U_H|\psi(0)\rangle \quad \text{where} \quad U_H = e^{-iHt/\hbar}$$

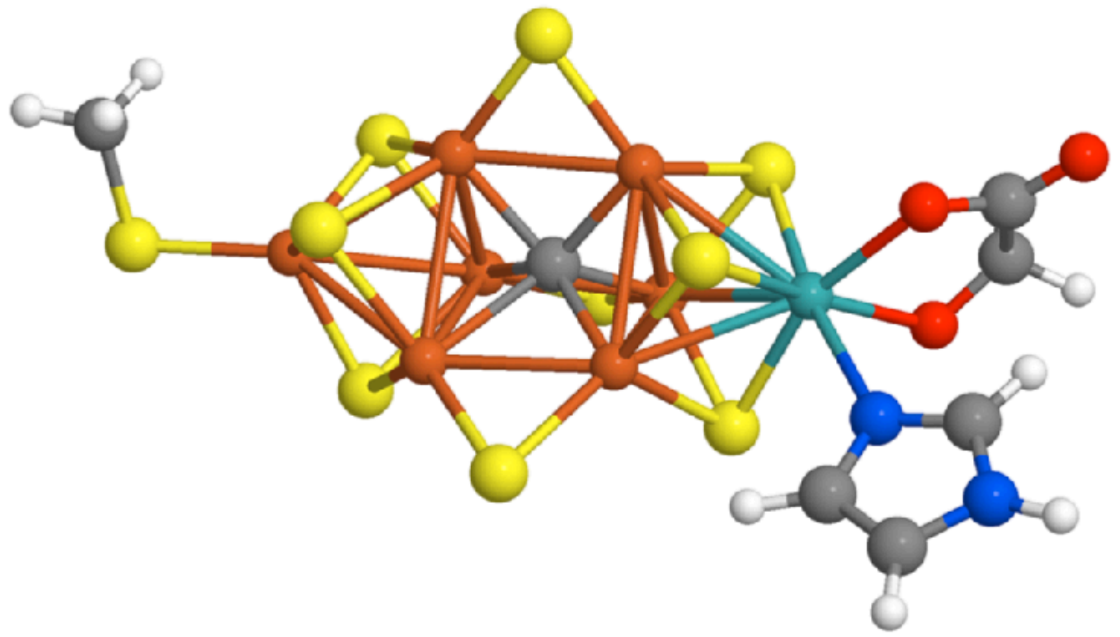
It is exponentially expensive to simulate an N-body quantum system on a classical computer: 2^N amplitudes!

- Cannot simulate more than $\mathcal{O}(10-100)$ particles

Quantum simulation

Feynman `81
Lloyd `96

A quantum computer can naturally simulate a quantum system



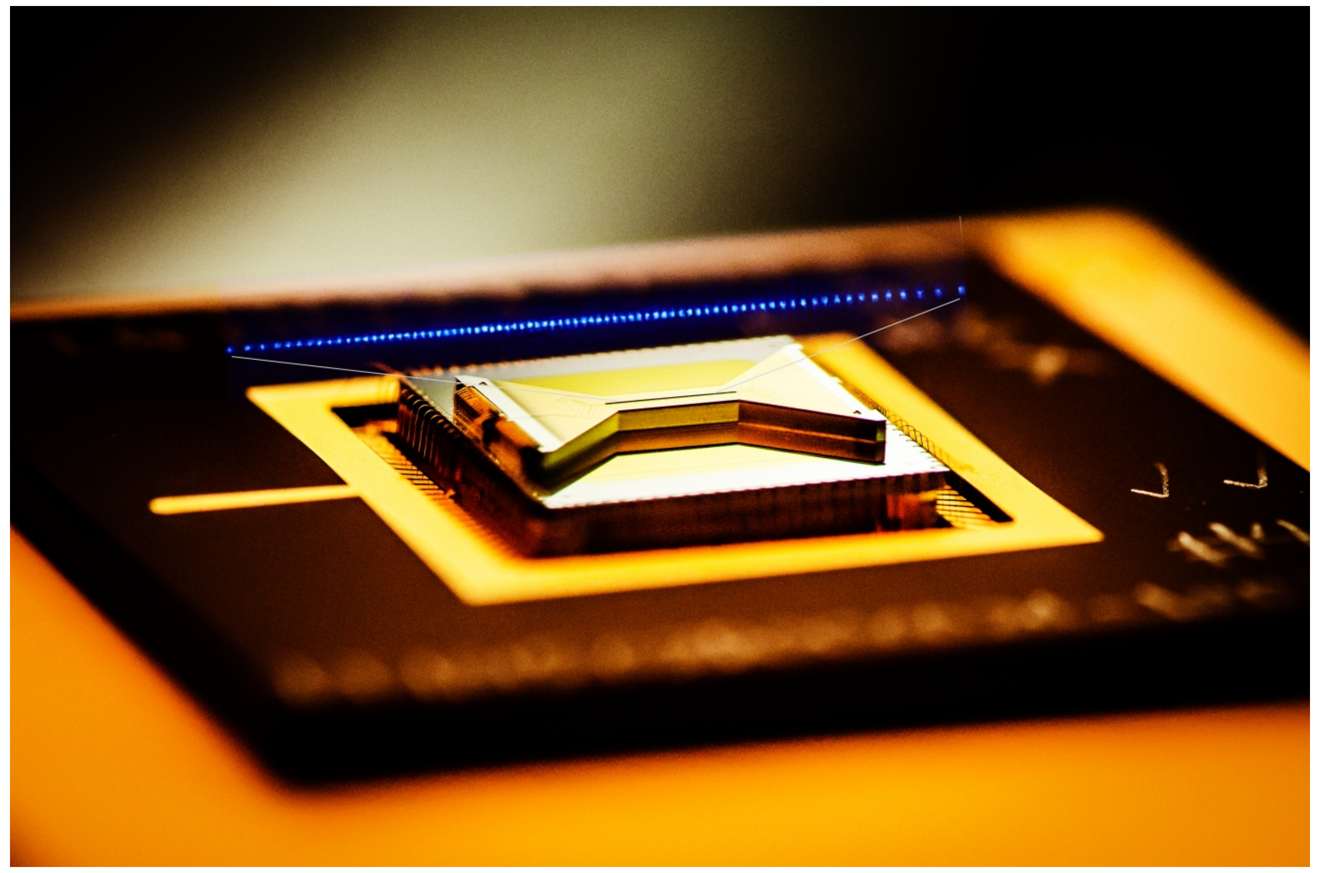
(1) Initial state preparation

$$|0\cdots0\rangle \rightarrow |\psi(0)\rangle$$

(2) Time evolution

$$|\psi(0)\rangle \;—\; \boxed{U_H(t)} \;—\; |\psi(t)\rangle$$

Need efficient encoding of U_H into quantum gates,
e.g. local interactions



IONQ

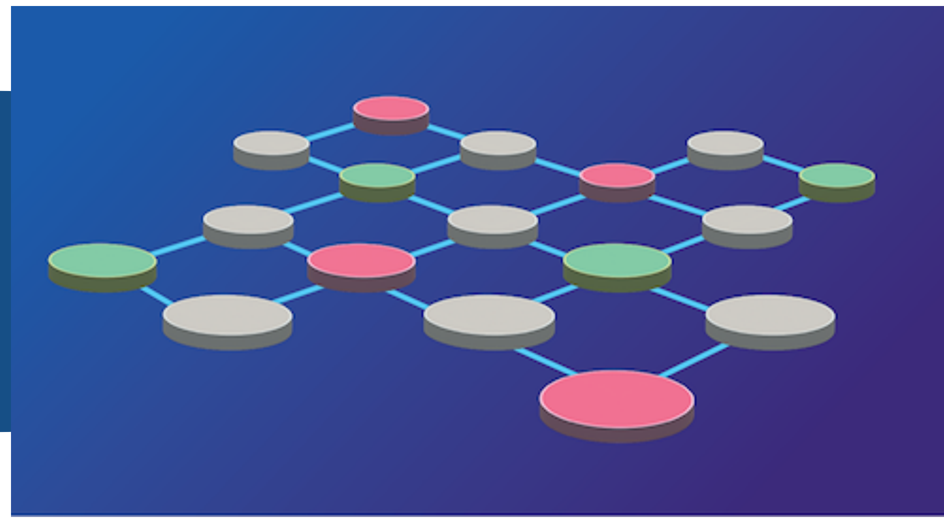
(3) Measurement

Outline

1. What is a quantum computer?

2. What are potential uses of quantum computing?

3. What are the challenges to achieve quantum advantage?



DiVincenzo Criteria

A quantum computer must satisfy the following:

- Scalable physical system with well-defined qubits
- Ability to initialize qubits
- Ability to measure qubits
- Universal set of quantum gates
- Qubit decoherence times much longer than gate latency

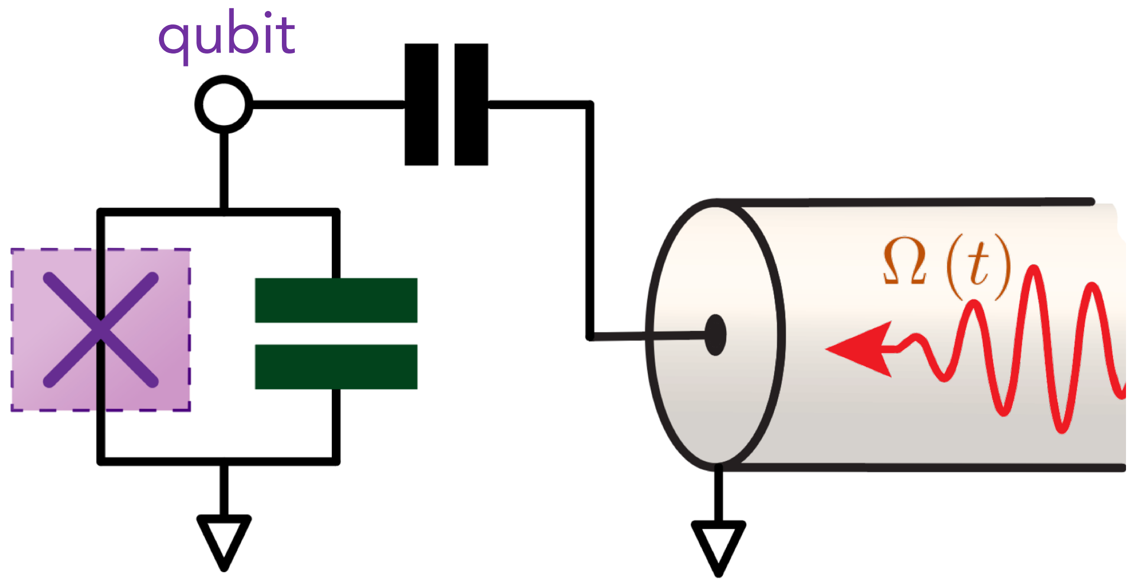
A variety of different physical systems are being explored, each with strengths

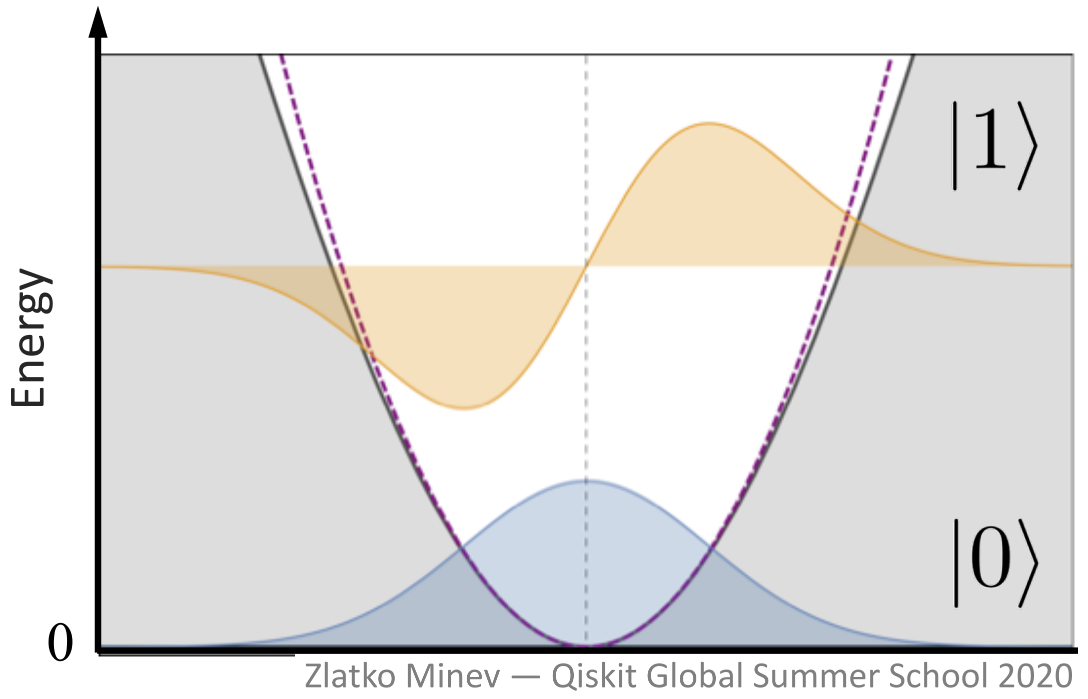
- Superconducting circuits
- Trapped ions
- Rydberg atoms
- Photonics
- Topological materials
- …

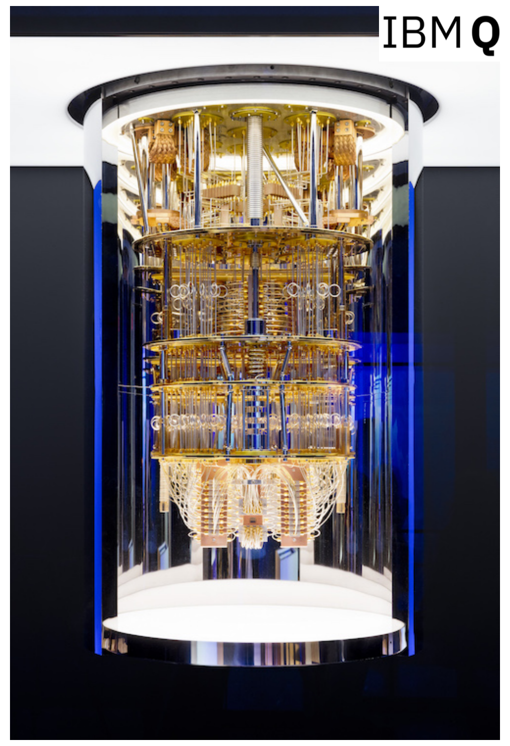
Quantum devices

Superconducting circuits

Google IBM Q rigetti



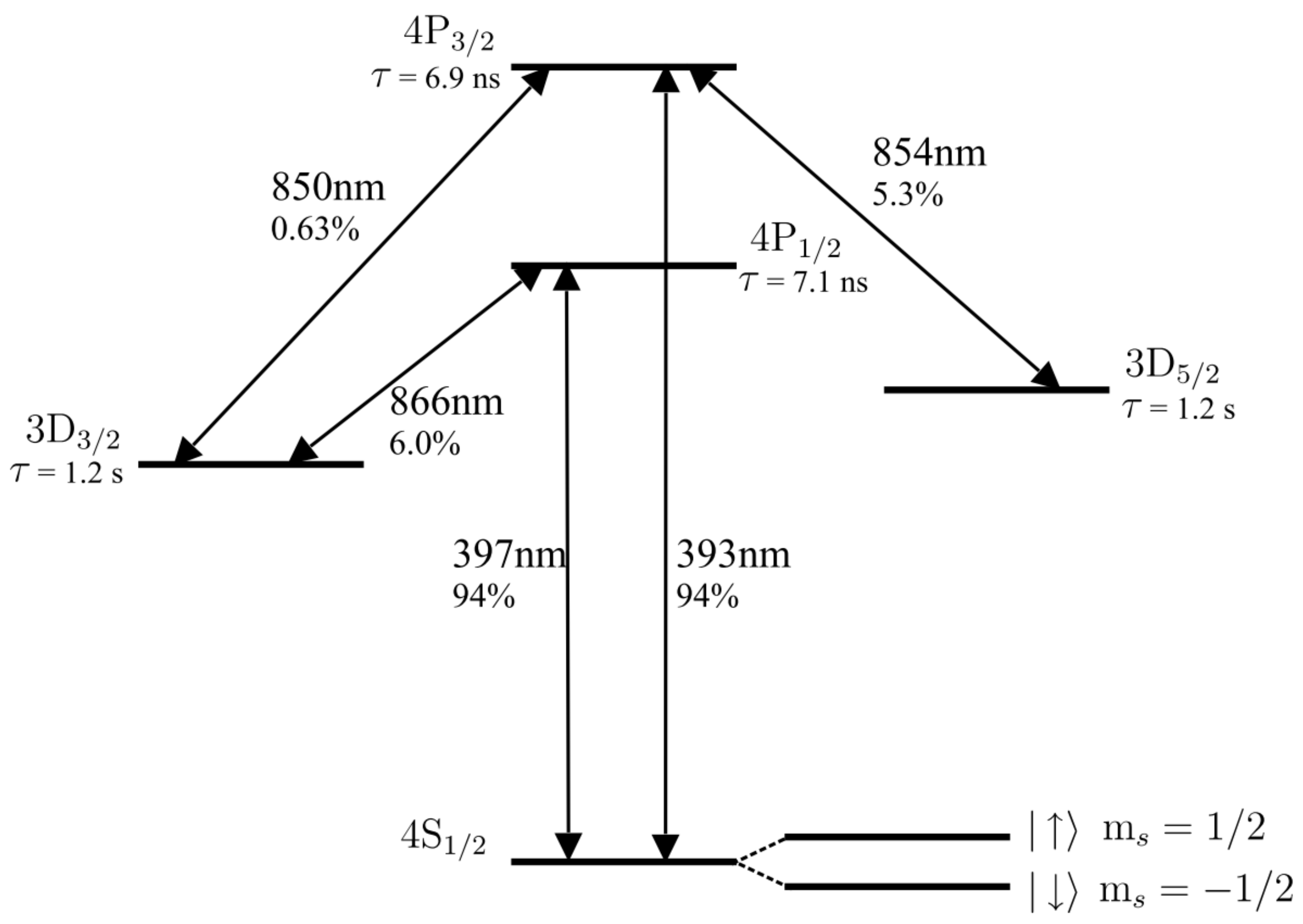


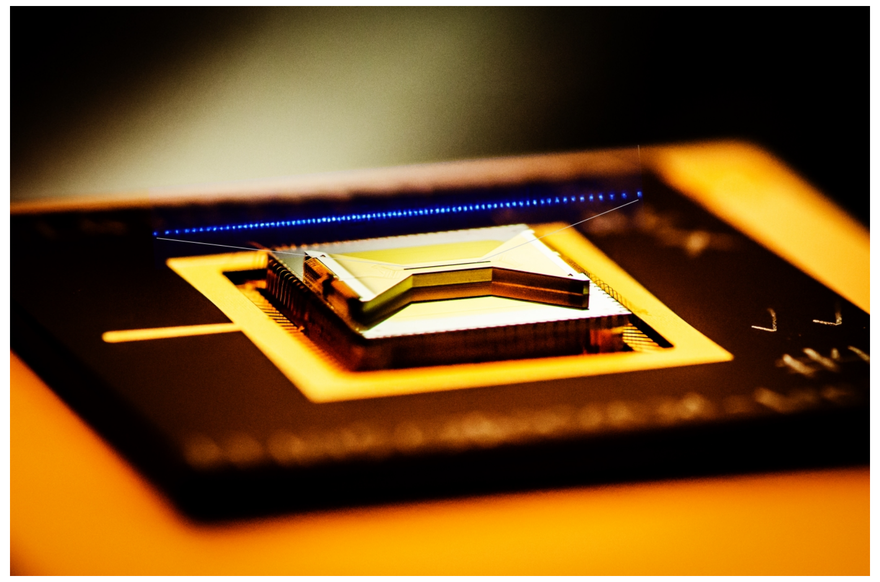


Qubits: Nonlinear quantum oscillator
Gates: Coupled microwave pulses

Trapped ions

IONQ QUANTINUUM





Qubits: Atomic energy levels (optical/hyperfine)
Gates: Coupled laser pulses

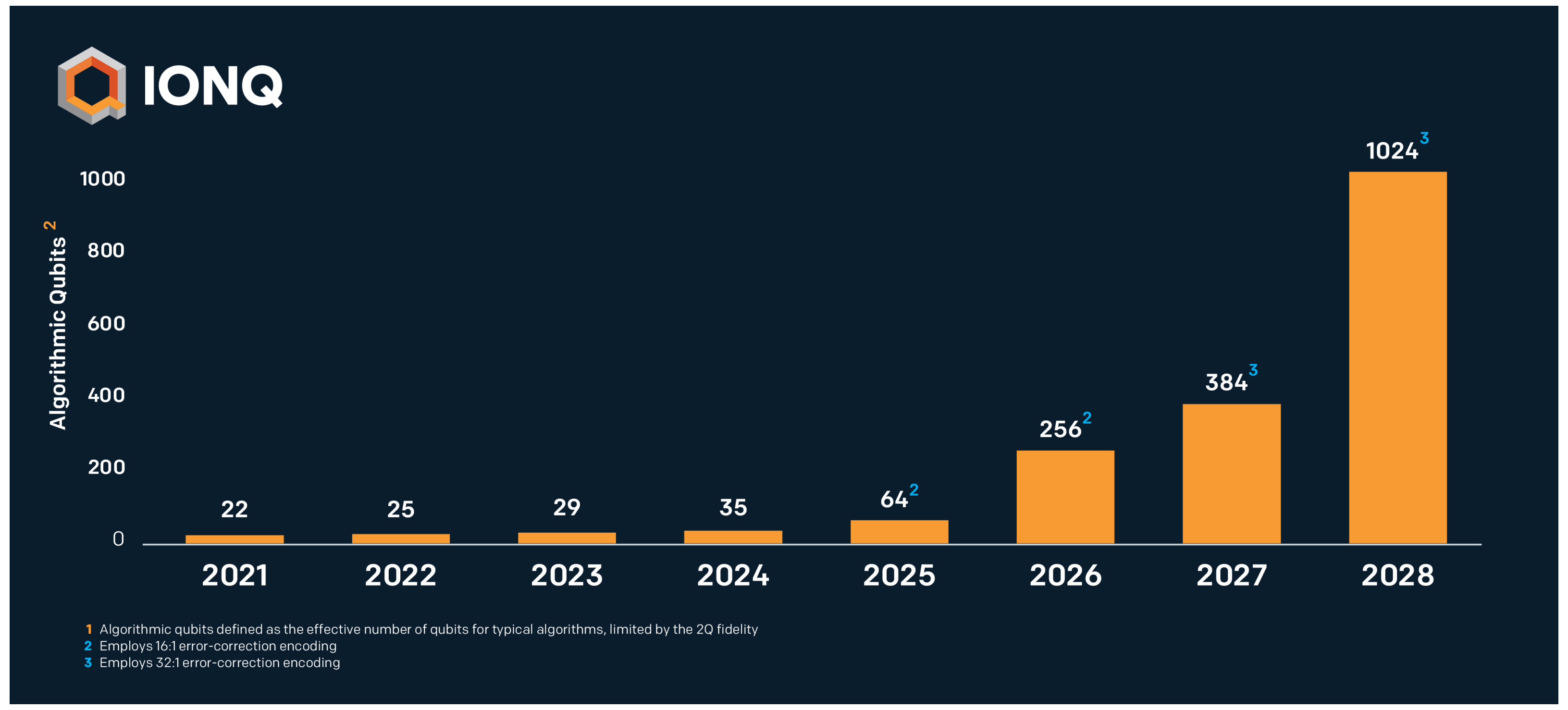
Rapid advances in qubit coherence times and quantum gates

State-of-the-art: $\mathcal{O}(10-100)$ qubits, $\mathcal{O}(100)$ two-qubit operations

Noisy quantum devices

Few qubits

Current devices are limited to $\mathcal{O}(10) - \mathcal{O}(100)$ qubits

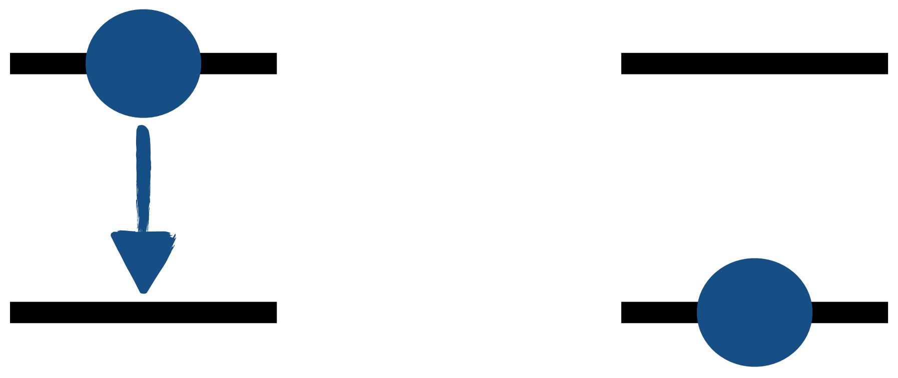


Need more qubits to achieve quantum advantage

Decoherence

The quantum state of a qubit is stable only for a limited time

T_1: decay time $|1\rangle \rightarrow |0\rangle$



T_2: dephasing time

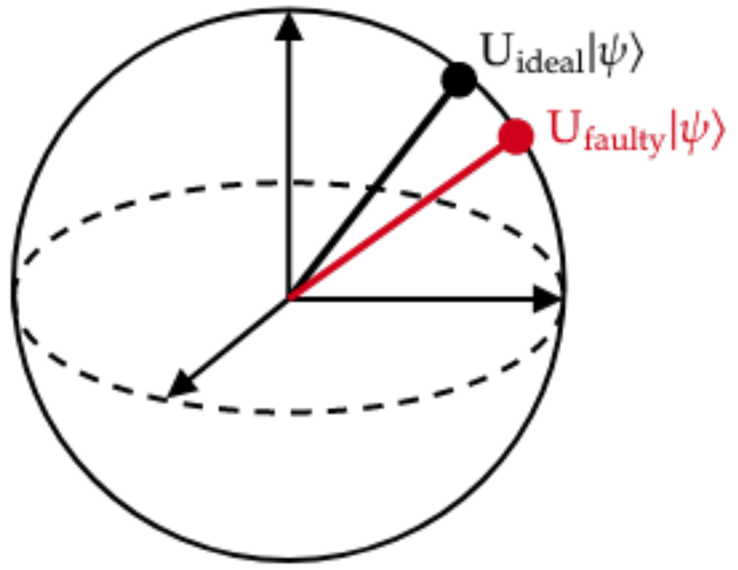
$$|1\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

Need longer coherence times to increase the "gate depth" of circuits

Gate noise

Single- and two-qubit gate operations are imperfect

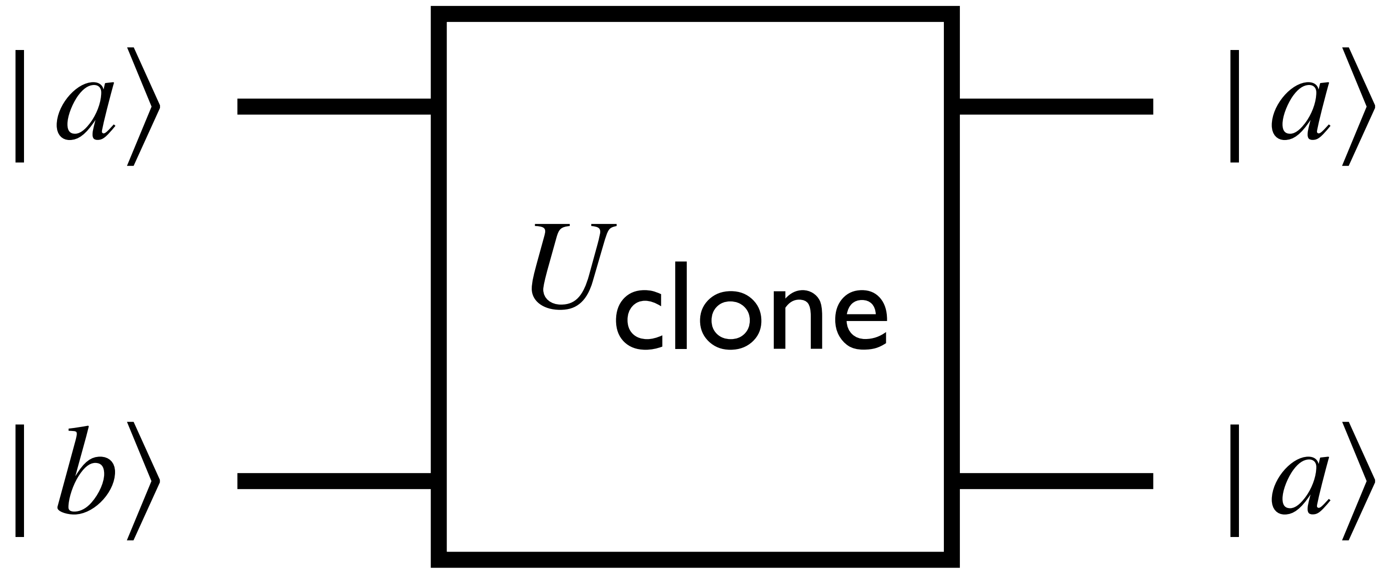
$$U_{\text{faulty}} = A \; U_{\text{ideal}}$$



Need smaller gate noise to perform quantum error correction

No cloning theorem

Can we copy arbitrary qubits, in order to provide redundancy?



Answer: No 😢

Proof: Suppose

$$U_{\text{clone}}\big(|a\rangle \otimes |b\rangle\big) = |a\rangle \otimes |a\rangle$$
$$U_{\text{clone}}\big(|a'\rangle \otimes |b\rangle\big) = |a'\rangle \otimes |a'\rangle\,.$$

Then taking inner product of both sides gives:

$$\langle a\,|\,a'\rangle = \langle a\,|\,a'\rangle^2$$

so $\langle a\,|\,a'\rangle = 0$ or 1 i.e. cannot copy arbitrary states.

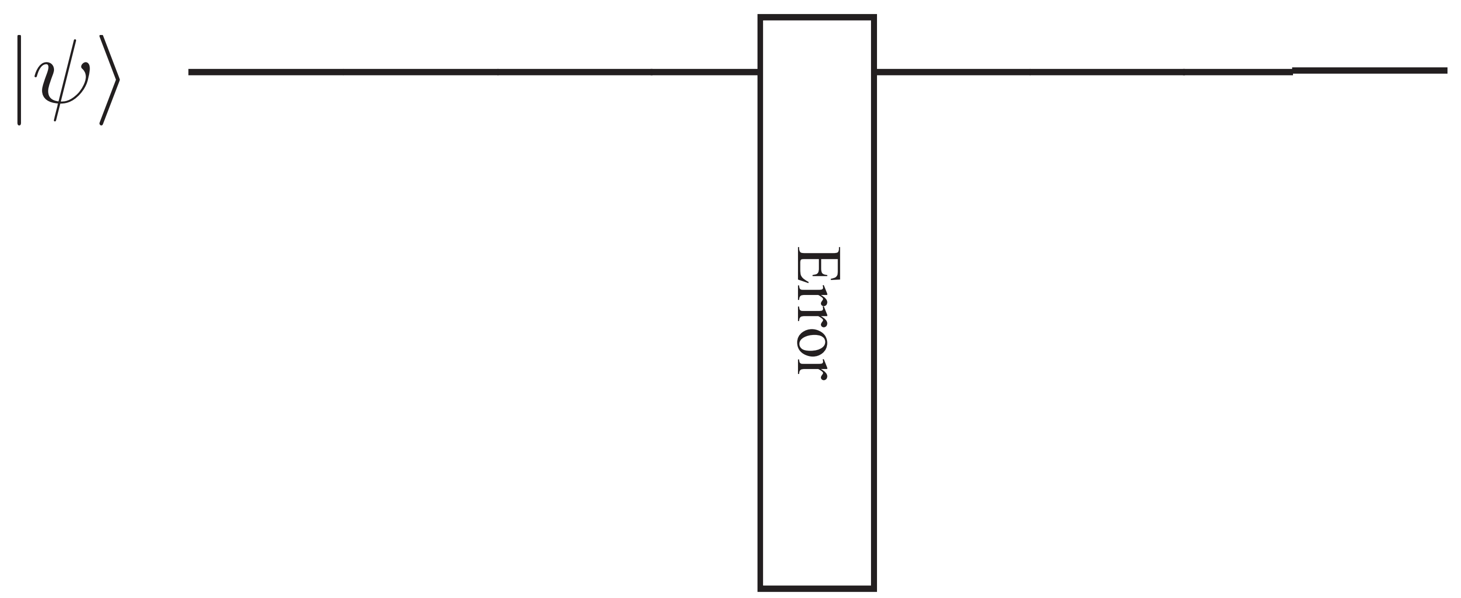
Quantum mechanics prevents us from using traditional error correction techniques

Quantum error correction

Idea: Encode one logical qubit in a larger set of physical qubits

Example: Bit flip code

- Suppose a bit flip (X gate) occurs with probability p



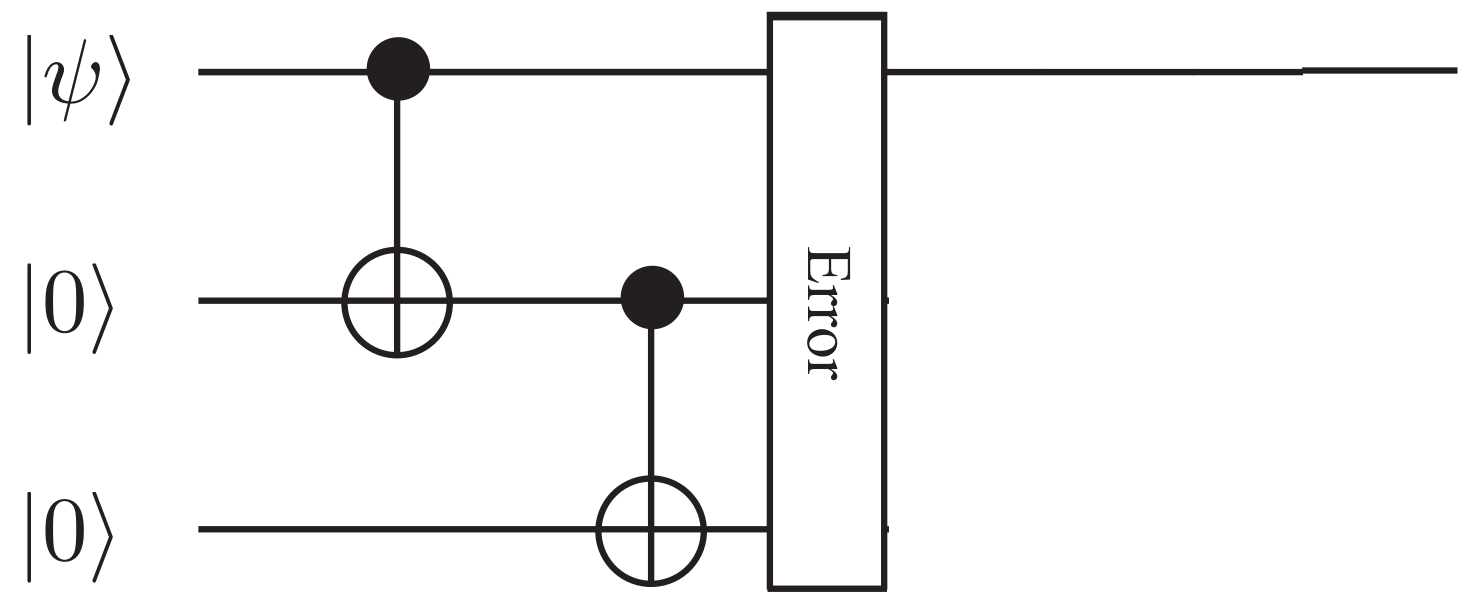
Devitt, Munro, Nemoto (2013)

Quantum error correction

Idea: Encode one logical qubit in a larger set of physical qubits

Example: Bit flip code

- Suppose a bit flip (X gate) occurs with probability p
- Encode our qubit $|\psi\rangle = a_0|0\rangle + a_1|1\rangle$ into three qubits: $|\psi_{\text{encoded}}\rangle = a_0|000\rangle + a_1|111\rangle$



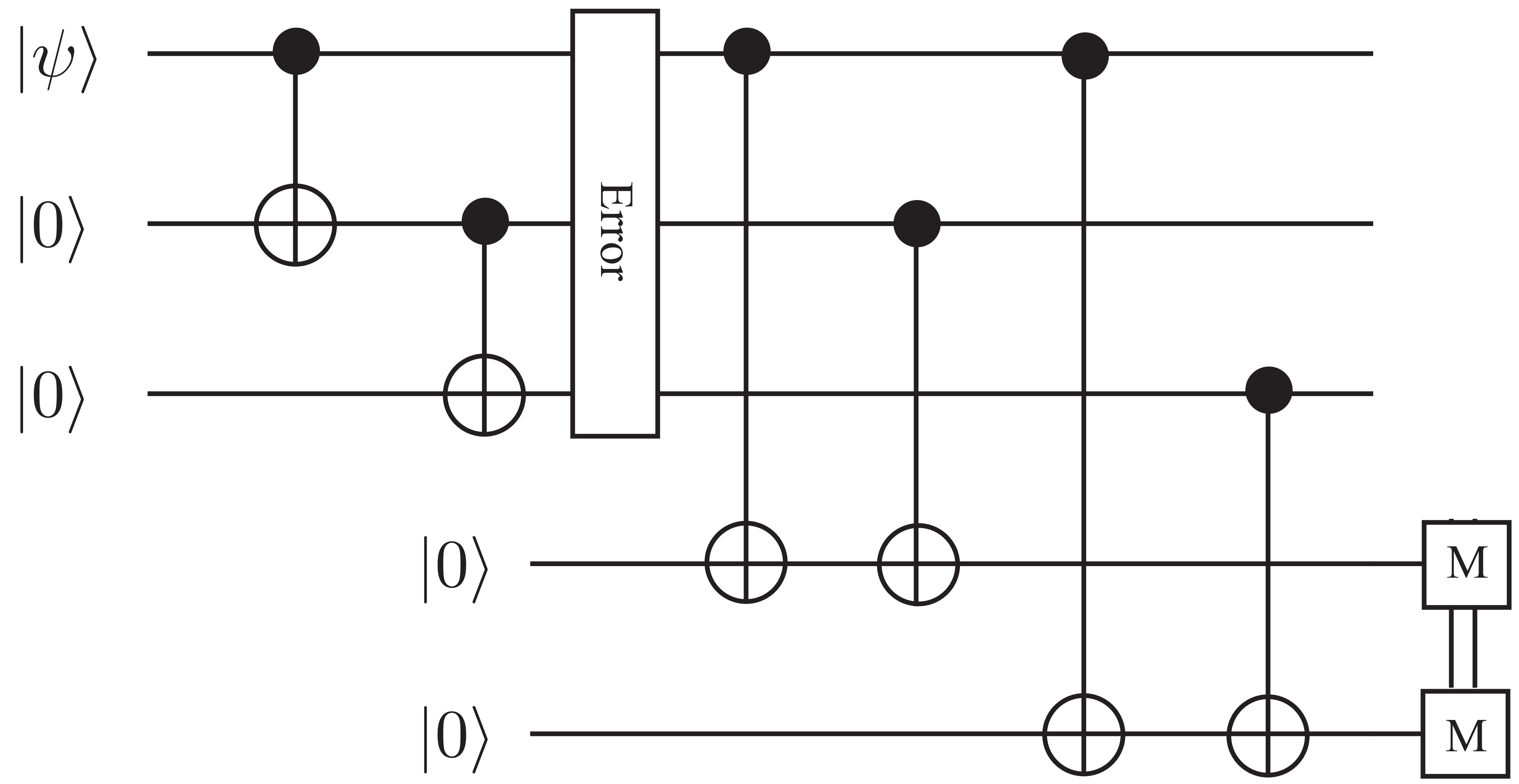
Devitt, Munro, Nemoto (2013)

Quantum error correction

Idea: Encode one logical qubit in a larger set of physical qubits

Example: Bit flip code

- Suppose a bit flip (X gate) occurs with probability p
- Encode our qubit $|\psi\rangle = a_0|0\rangle + a_1|1\rangle$ into three qubits: $|\psi_{\text{encoded}}\rangle = a_0|000\rangle + a_1|111\rangle$
- Introduce ancilla qubits to measure the parity of the three qubits



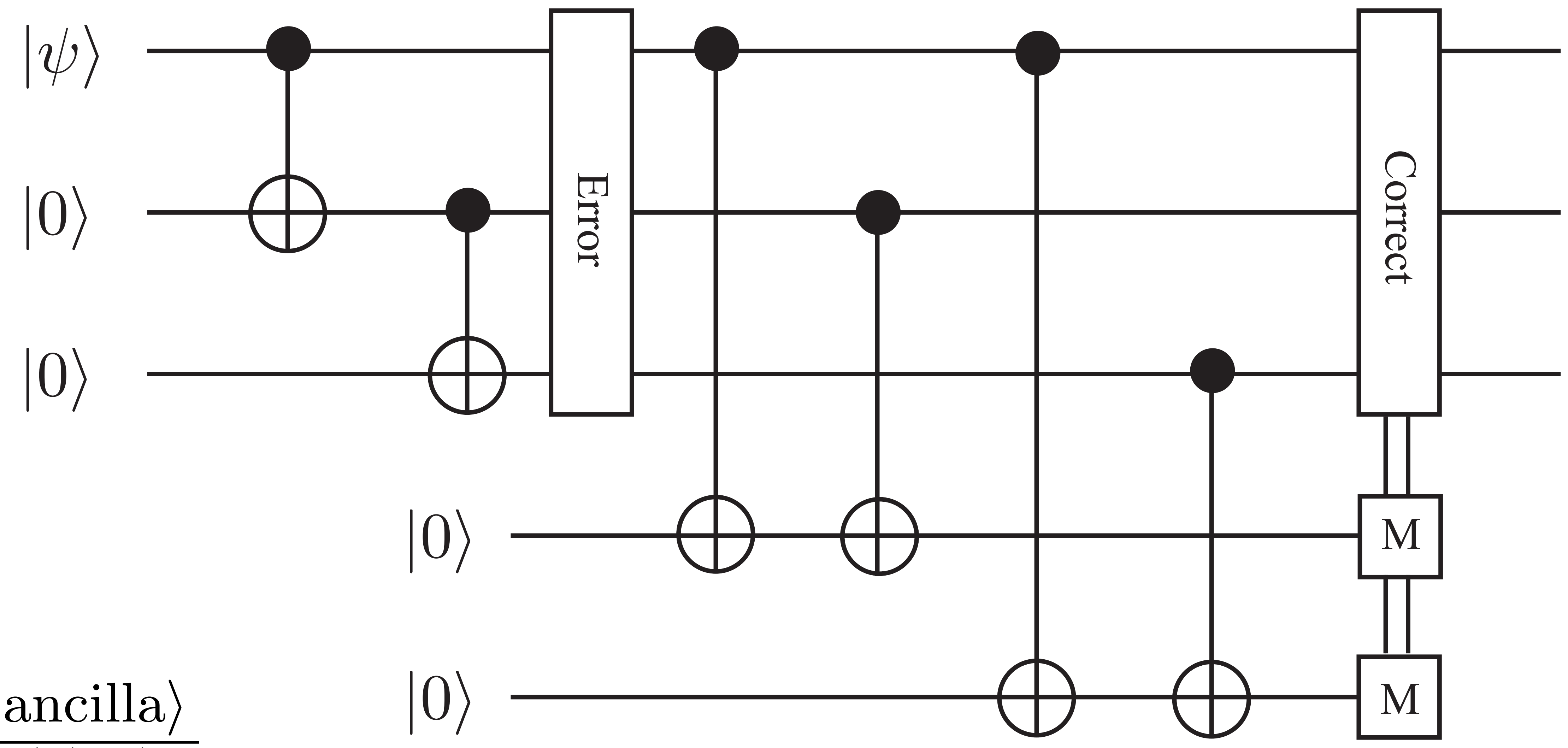
Devitt, Munro, Nemoto (2013)

Quantum error correction

Idea: Encode one logical qubit in a larger set of physical qubits

Example: Bit flip code

- Suppose a bit flip (X gate) occurs with probability p
- Encode our qubit $|\psi\rangle = a_0|0\rangle + a_1|1\rangle$ into three qubits: $|\psi_{\mathrm{encoded}}\rangle = a_0|000\rangle + a_1|111\rangle$
- Introduce ancilla qubits to measure the parity of the three qubits
- Perform correction



Error Location	Final State, $|\mathrm{data}\rangle\,|\mathrm{ancilla}\rangle$
No Error	$\alpha\,|000\rangle\,|00\rangle + \beta\,|111\rangle\,|00\rangle$
Qubit 1	$\alpha\,|100\rangle\,|11\rangle + \beta\,|011\rangle\,|11\rangle$
Qubit 2	$\alpha\,|010\rangle\,|10\rangle + \beta\,|101\rangle\,|10\rangle$
Qubit 3	$\alpha\,|001\rangle\,|01\rangle + \beta\,|110\rangle\,|01\rangle$

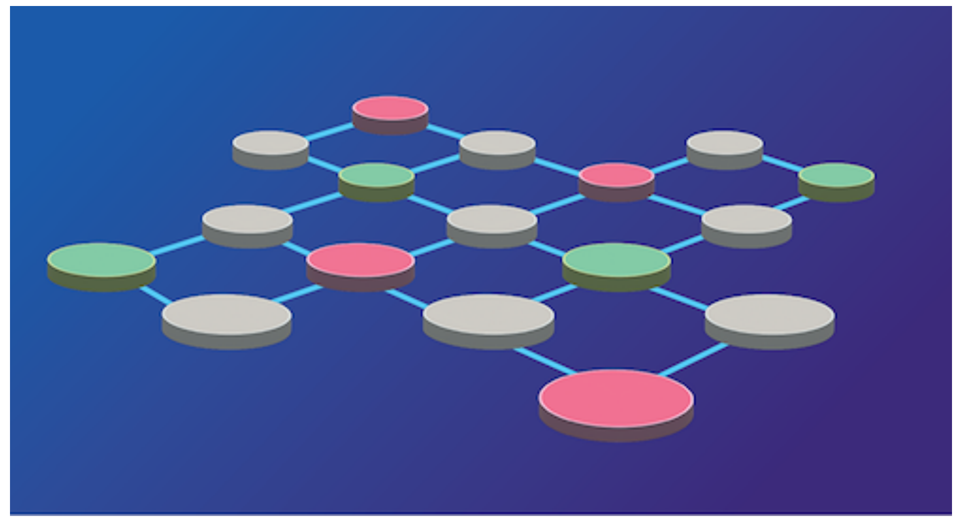
Devitt, Munro, Nemoto (2013)

Quantum error correction

Idea: Encode one logical qubit in a larger set of physical qubits

There are a variety of error correction codes:

- Shor code
- Steane code
- Surface codes
- …

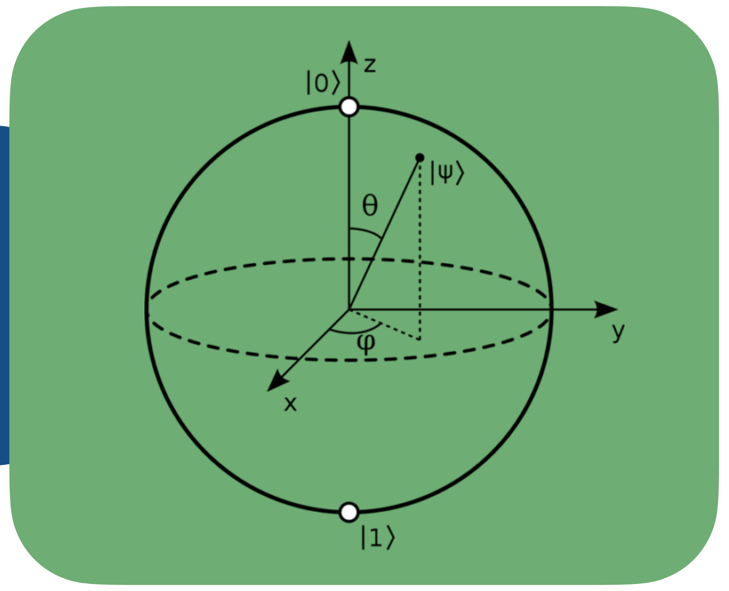


Quantum threshold theorem: If errors are below a certain threshold, then you can correct errors faster than you introduce them

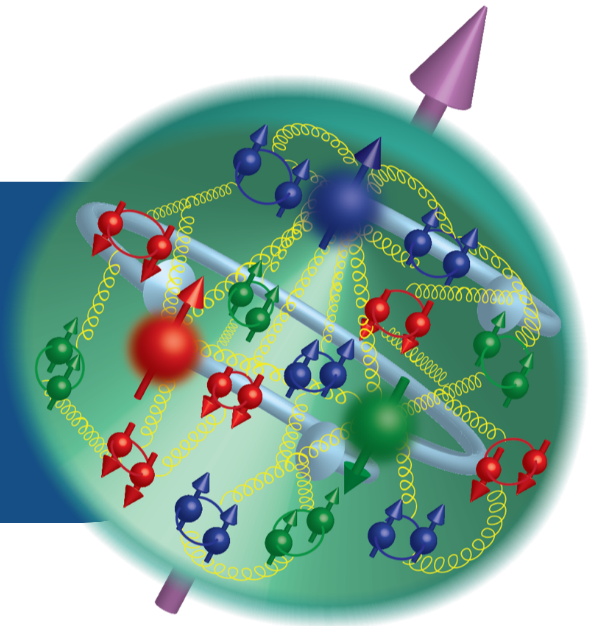
→ Demonstrating "break-even" point is active goal of research

Outline

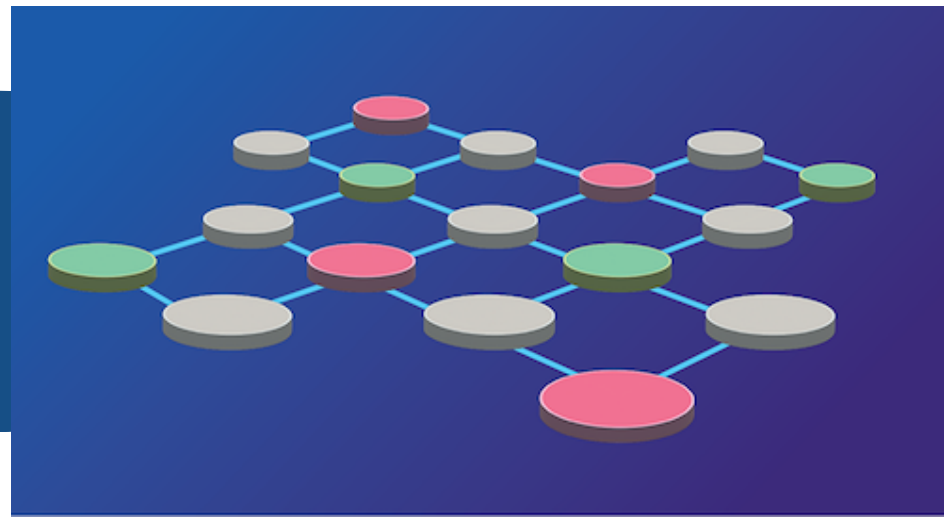
1. What is a quantum computer?



2. What are potential uses of quantum computing?



3. What are the challenges to achieve quantum advantage?



Additional reading material

Not required, but you may find them useful and interesting!

Basics:

- Introduction to qubits
- Unitary gates: Nielsen+Chuang Section 1.3

Introduction to complexity classes

Quantum computing overviews:

- https://arxiv.org/pdf/1905.07240.pdf
- https://arxiv.org/pdf/1801.00862.pdf

Quantum simulation: https://arxiv.org/pdf/1907.03505.pdf

Qiskit tutorials from IBM Quantum Lab

- Create an account: https://quantum-computing.ibm.com/lab
- Login and navigate to qiskit-tutorials/qiskit/circuits
- Complete the following two tutorials:

  01_circuit_basics.ipynb

  3_summary_of_quantum_operations.ipynb
- Navigate to qiskit-tutorials/qiskit/simulators
- Complete the following two tutorials:

  2_device_noise_simulation.ipynb

  3_building_moise_models.ipynb
- Navigate to qiskit-tutorials/qiskit-ibm-runtime
- Complete the following tutorial:

  01_introduction_ibm_cloud_runtime.ipynb