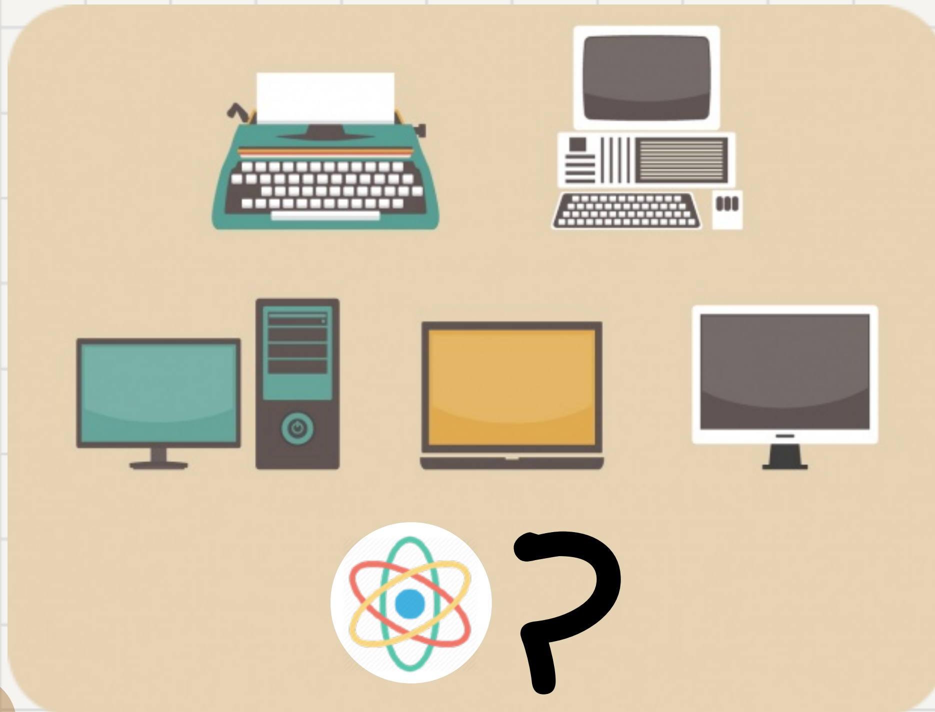
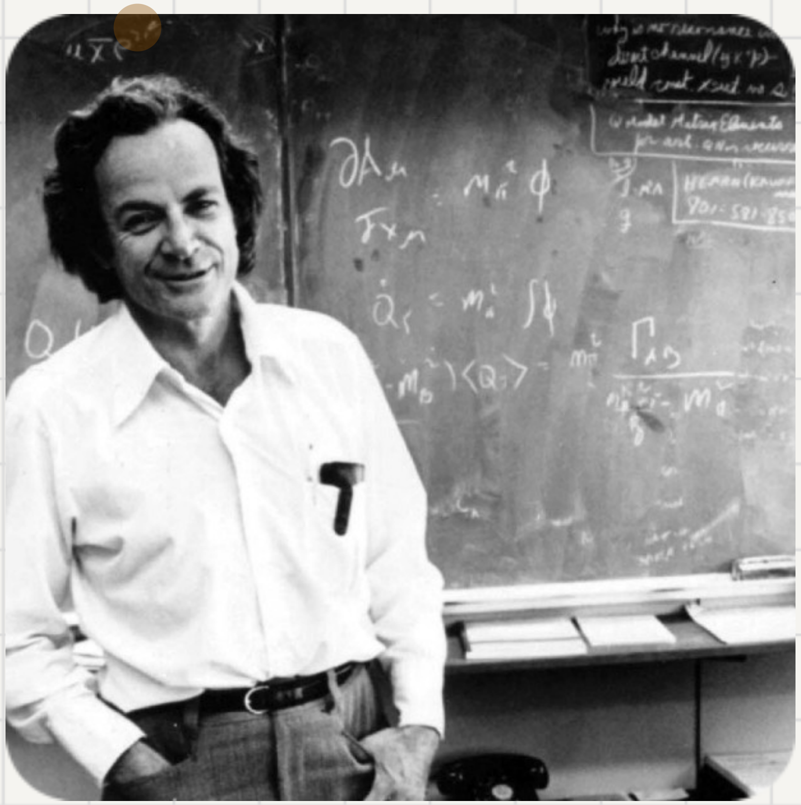
Complexity Theory

Lecture 12

Quantum Complexity

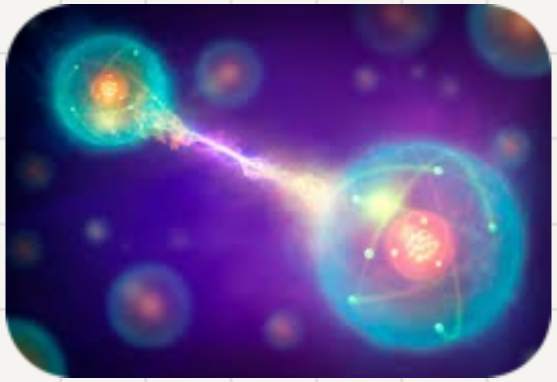
What is quantum computing



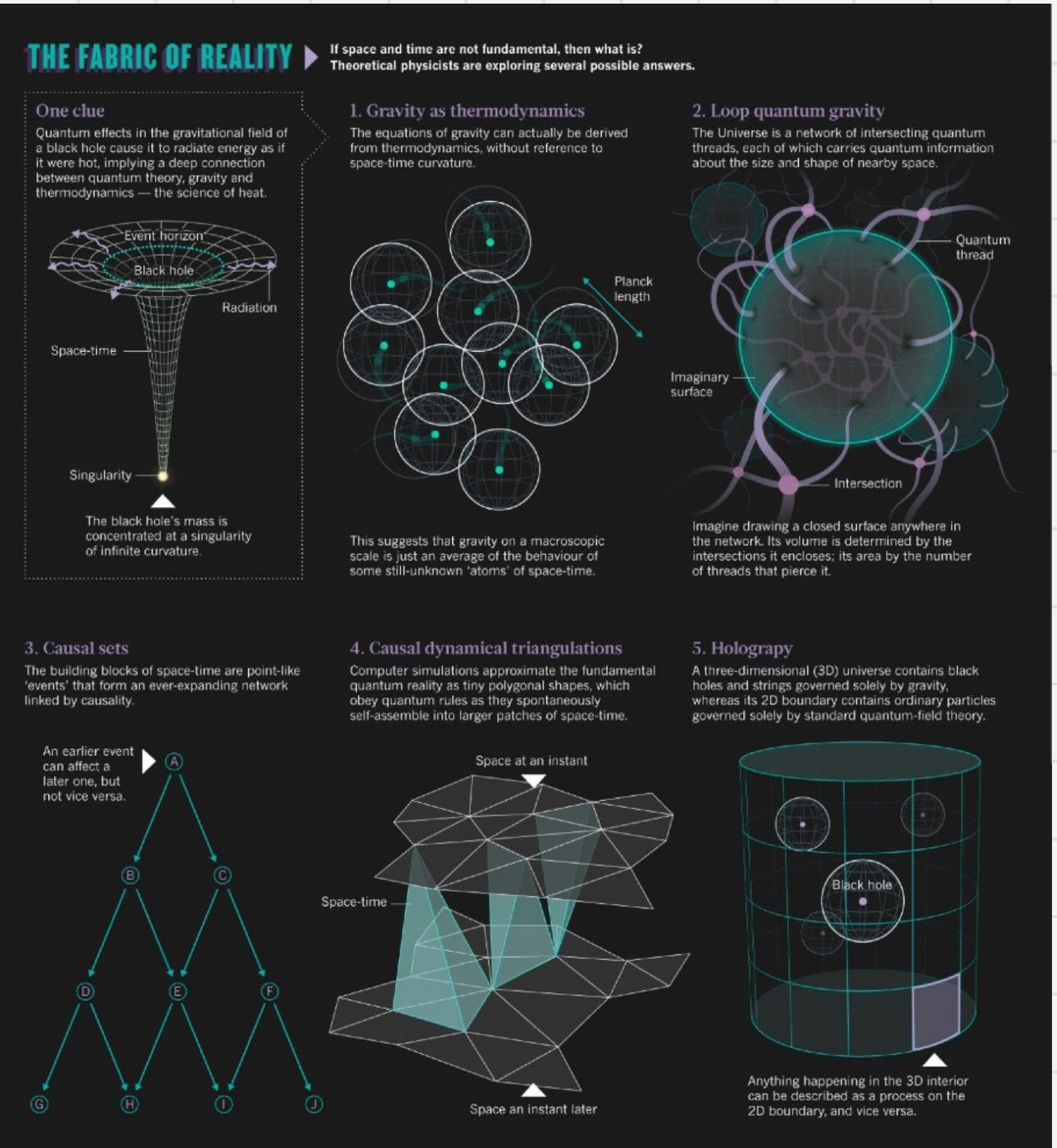


The big idea:

Computers that rely on the Powerful, but unintuitive, principles of quantum mechanics



Quantum computing is... nothing less than a distinctively new way of harnessing nature... it will be the first technology that allows useful tasks to be performed in collaboration between parallel universes.



—Fabric of reality

David Deutsch

"if you teach an introductory course on quantum mechanics, and the students don't have nightmares for weeks, tear their hair out, wander around with bloodshot eyes, etc., then you probably didn't get the point across."

Time travel

Parallel universes

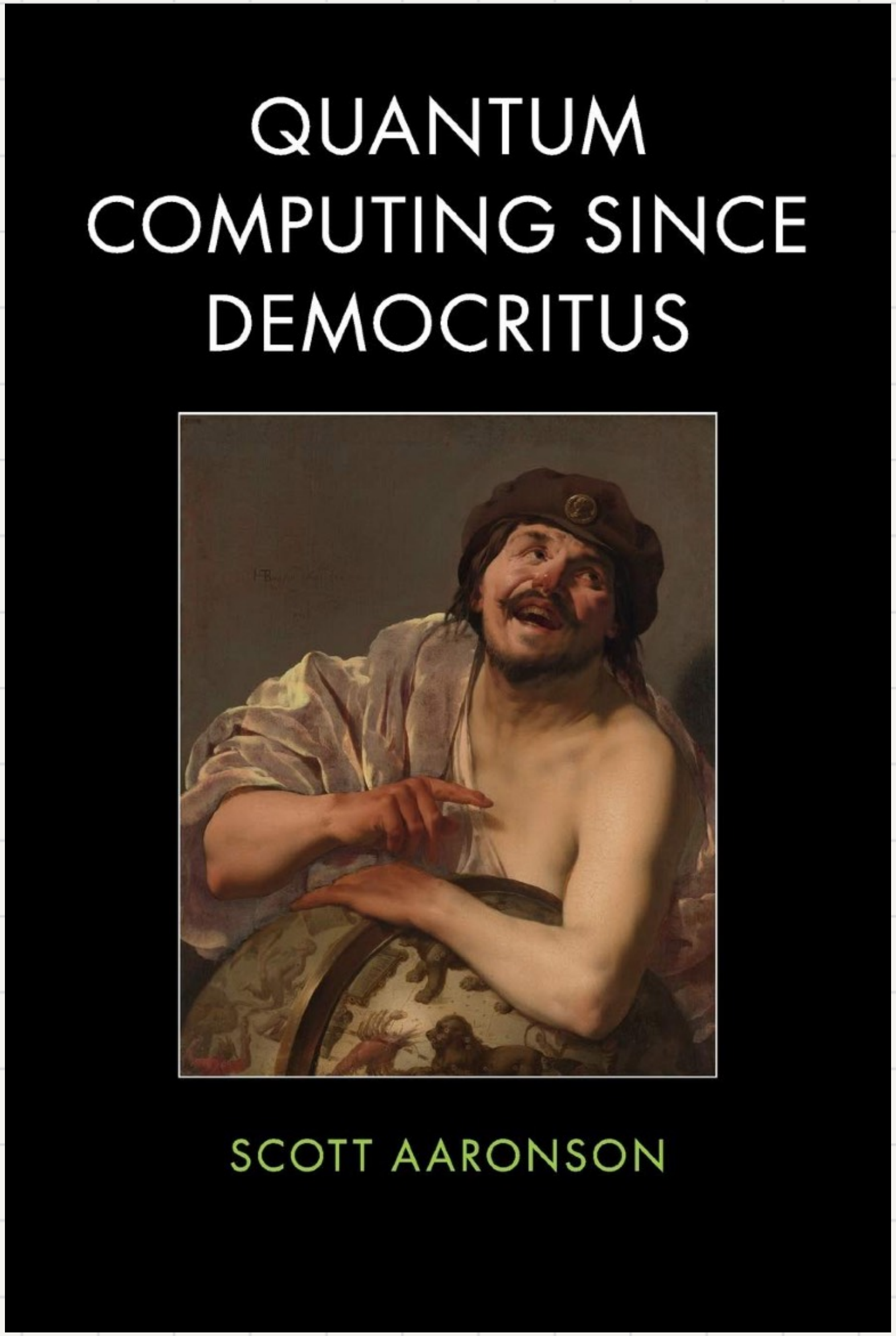
Teleportation

Cloning

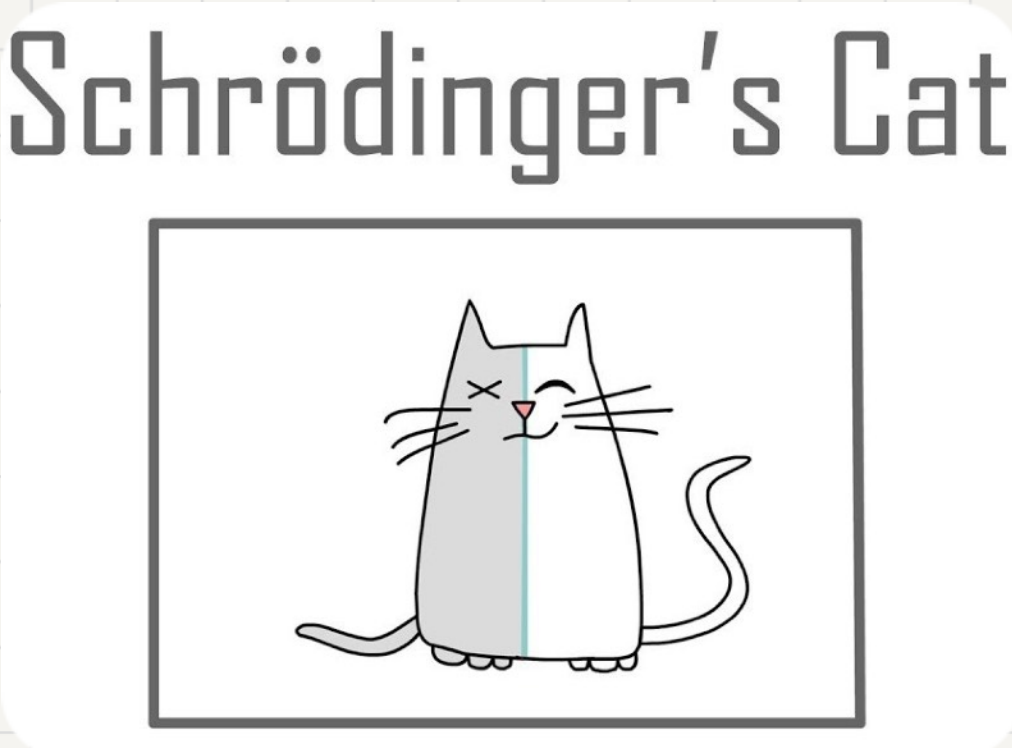
Faster-than-light communication

Determinism

Chaos



Superposition, parallel worlds, and cats

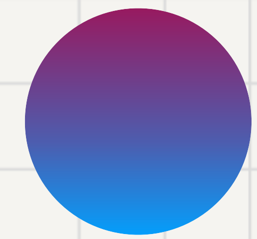


Bit

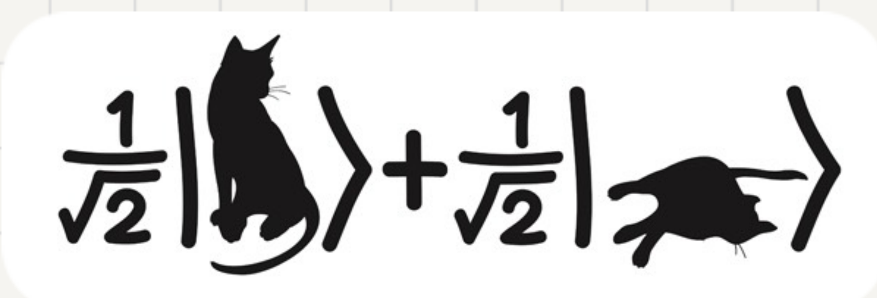
1

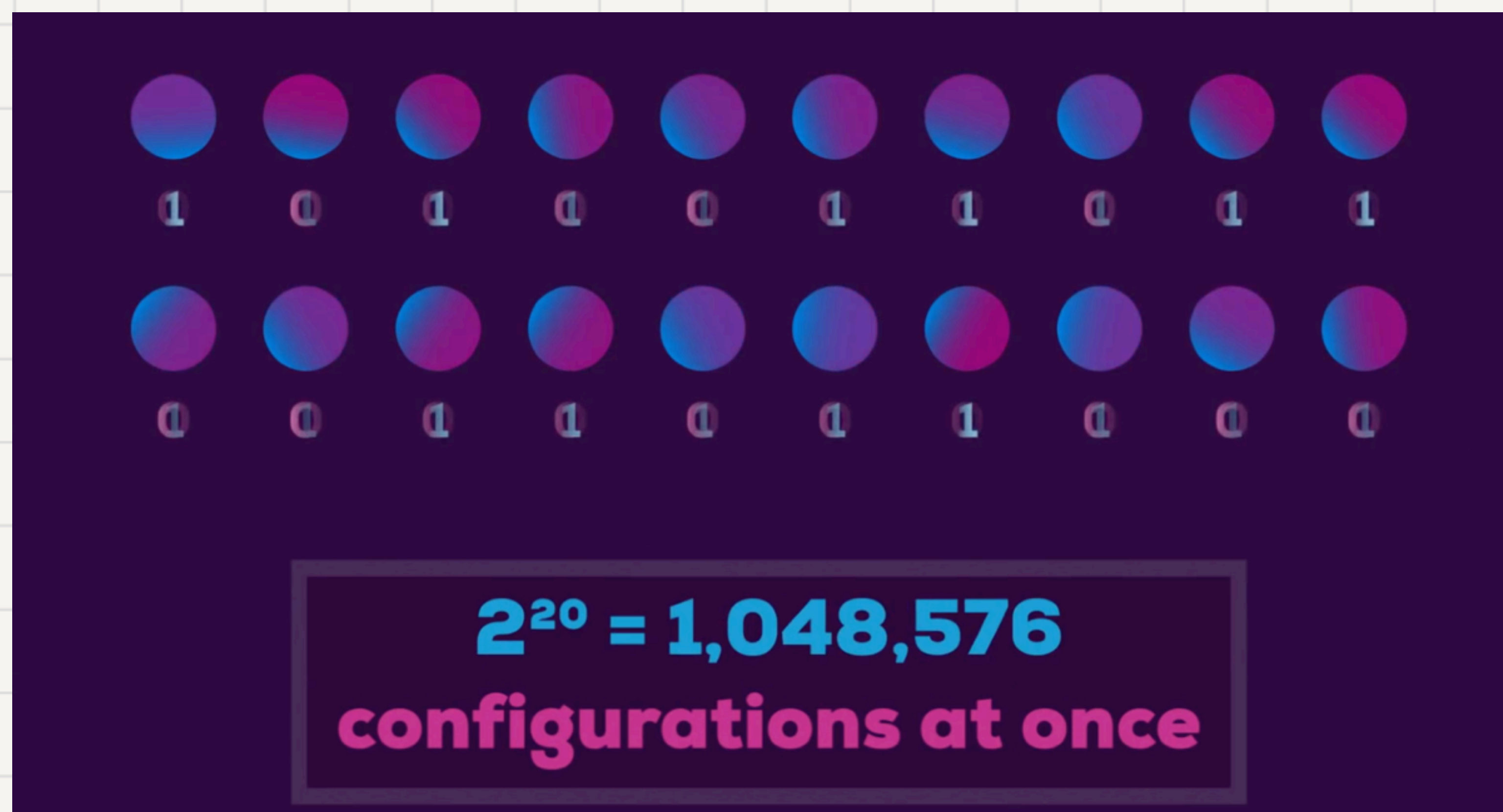
0

Qubit



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





A crash course on quantum computing

Quantum mechanics is a $\mathbb{C}$ probability theory

Fully captured by 4 postulates:

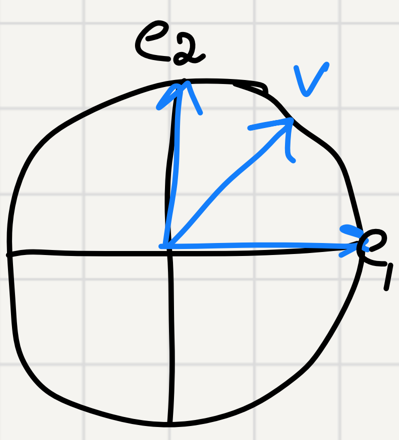
(I) Superposition

A quantum state is a vector $v \in \mathbb{C}^n$ s.t. $\|v\|_2^2 = 1$.

Ex: a qubit $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$, $|\alpha|^2 + |\beta|^2 = 1$

(II) Measurement

Measuring state $(\alpha_1, \dots, \alpha_n)$ in the computational basis collapses it to e_i w.p. $|\alpha_i|^2$



Mathematical abstraction of quantum computing

III Evolution

A quantum state $v \in \mathbb{C}^n$ evolves to $v' \in \mathbb{C}$ via a unitary map

$$\begin{pmatrix} | \\ v' \\ | \end{pmatrix} = \begin{pmatrix} \\ U \\ \end{pmatrix} \cdot \begin{pmatrix} | \\ v \\ | \end{pmatrix}$$

IV Entanglement

States $u, v \in \mathbb{C}^n$ are composed via the tensor product $u \otimes v$.

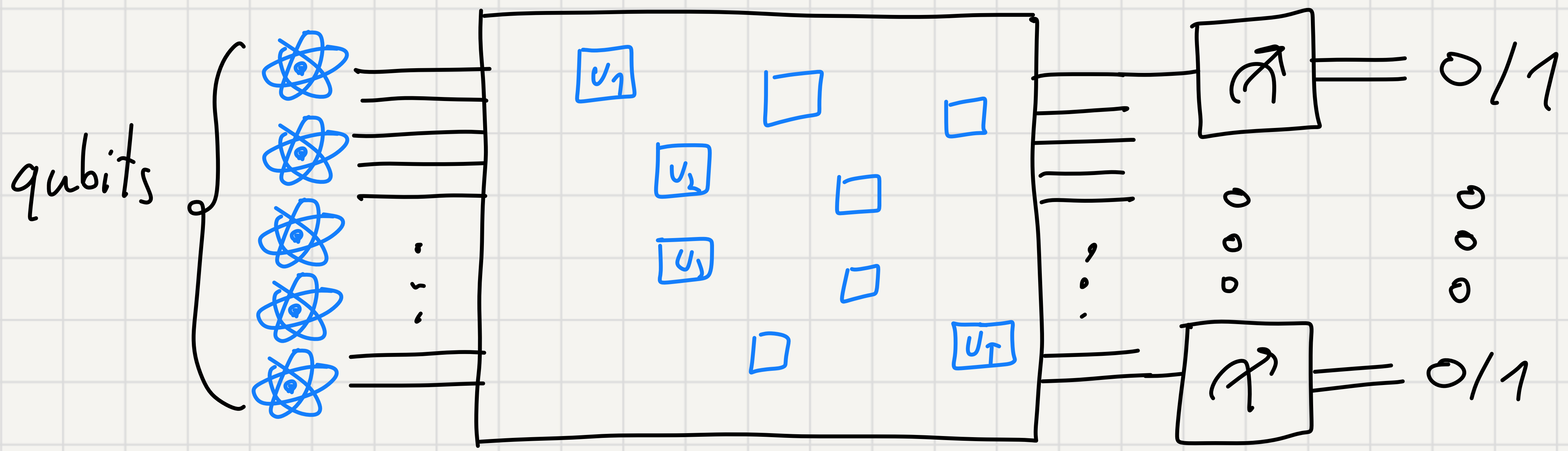
Ex:
$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{array}{c c} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} & \begin{array}{c} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{array} \end{array}$$

Entangled states are <u>not</u> of that form!

Quantum algorithms

Quantum Turing machines ...are not very nice to work with...

Instead, we typically work with quantum circuits.



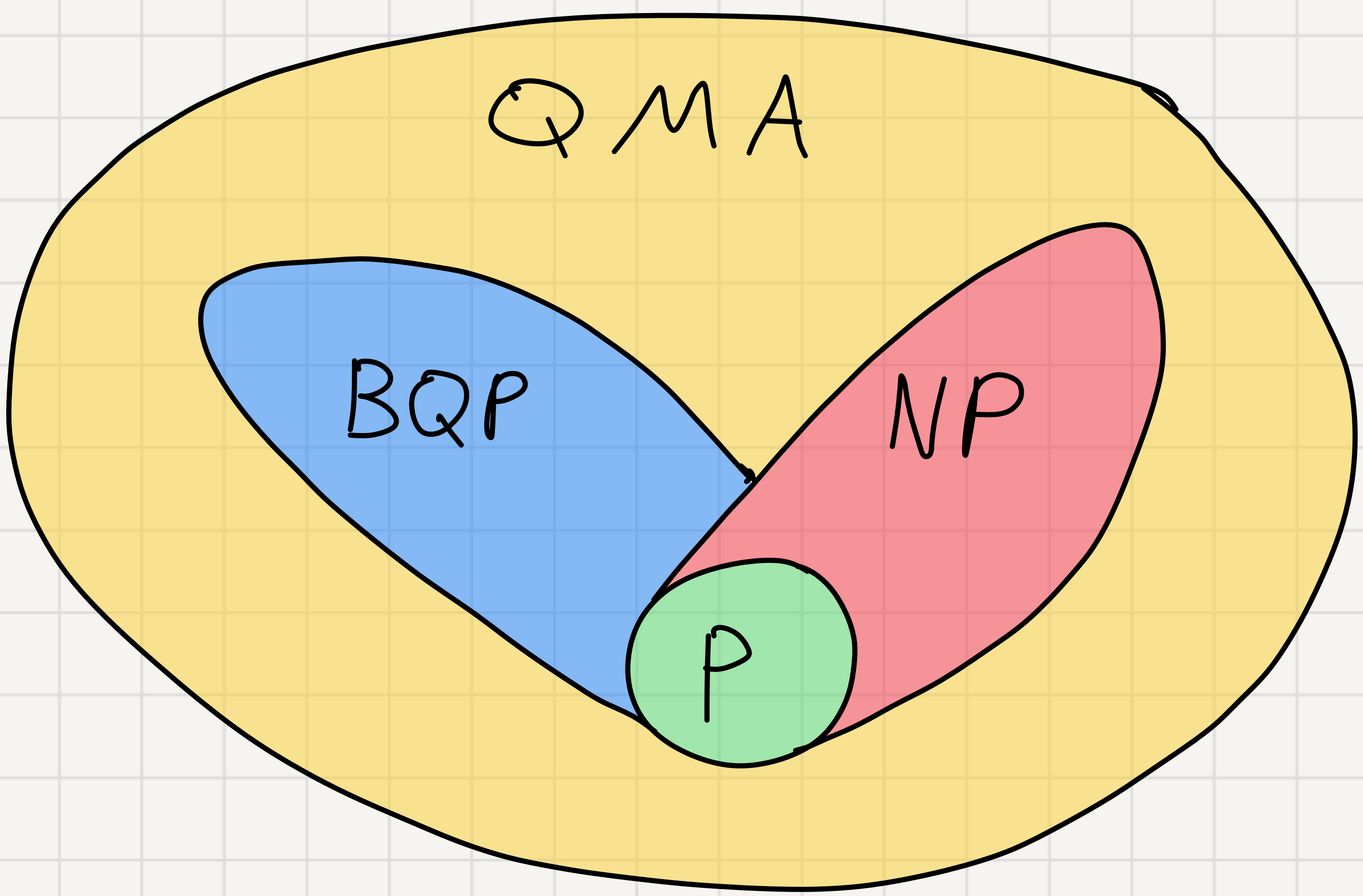
We apply quantum gates (unitary maps) to qubits.

Measure at the end to get classical bits.

Quantum complexity classes

BQP = "quantum P"

QMA = "quantum NP"



BQP

The set of all problems solvable by a poly-time uniform quantum circuits $(C_n)_{n\in\mathbb{N}}$ of polynomial size, w.p. $\geq 2/3$

(i.e., $\forall_{x\in\{0,1\}^n}$ $\Pr[C_n(x) = \mathbb{1}_L(x)] \geq 2/3$)

$T(n)$-uniformity: C_n can be generated in $T(n)$ time

Circuit size: # gates → time complexity

Error reduction

Let A be a q-algo for computing f such that $\Pr[A(x) = f(x)] \geq 2/3 \quad \forall x$.

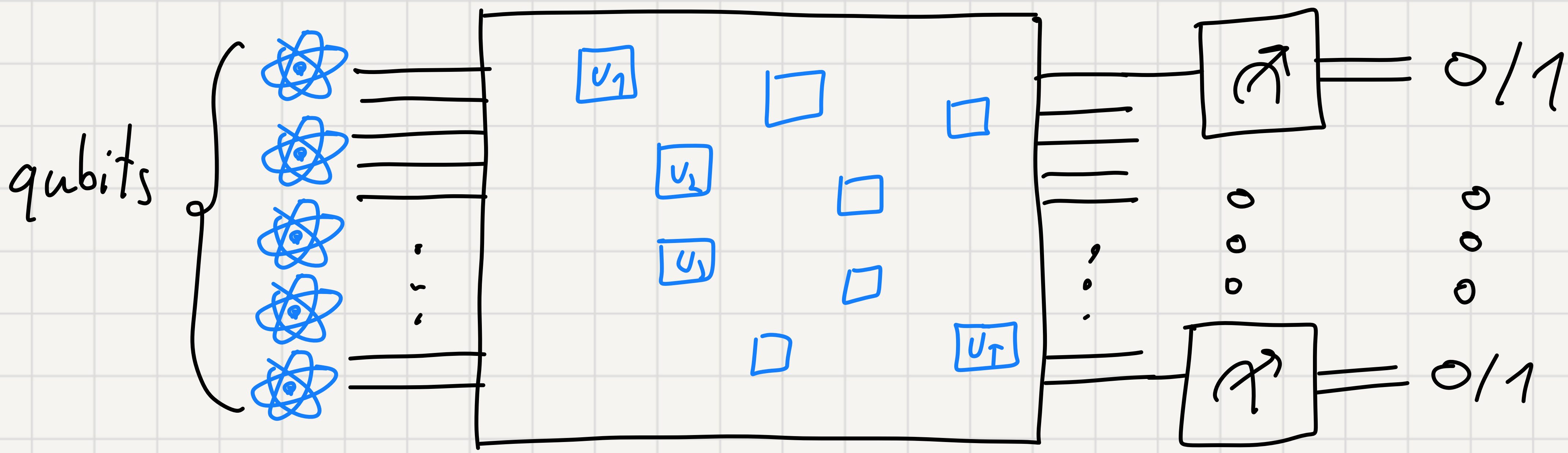
$1/3$ error prob. can be reduced to ε!

Repeat A: $A_1, A_2, \dots, A_{\underbrace{\log(\frac{1}{\varepsilon})}_{t}}$, rule by Maj.

Chernoff bound:

$$\Pr\left[\left|\frac{\sum_{i\in[t]} A_i}{t} - \mathbb{E}[X_i]\right| \leq \frac{1}{6}\right] \leq \exp(-t)$$

Quantum algorithms



3 examples where quantum algorithms excel:

(I) Finding sub-group structure (Shor's factoring)

(II) Rapid mixing of Markov chains (Grover's search)

(III) Computing Fourier Transforms (QFT)

Factoring

Given $n \in \mathbb{N}$, output primes $p_1, \dots, p_n$ s.t.

$n = p_1 \cdot p_2 \cdot \dots \cdot p_n$

Decision problem $\text{Factor}(n, k) = 1$

iff n has a prime factor $\leq k$

Shor's algorithm $\text{Factor} \in BQP$

We know $\text{Factor} \in NP \cap coNP$.

Grover's search

Given a string $x \in \{0,1\}^n$, output $i \in [n]$ such that $x_i = 1$

0	0	0	0	0	0	0	0	0	1	0	0	0	0

Classical complexity? $\Omega(n)$ Quantum complexity $\Omega(\sqrt{n})$

Quantum Fourier Transform

Given $(f_1, f_2, \ldots, f_N) \in \mathbb{C}^N$, output the DFT $(\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_N)$

Classical complexity? $O(N \log N)$ Quantum complexity $\tilde{O}(\log N)$

Ask Me Anything