Part IV

The BB84 protocol: Alice and Bob's hotline

One-time pad: a Catch-22 situation

- Alice wants to send Bob a message m without Eve being able to learn anything about its content
- This can be achieved if Alice and Bob share in advance a string k of random bits:
  - Alice computes $x = m \oplus k$ and sends x to Bob
  - Eve cannot learn anything from x ($Pr(M = m|X = x) = Pr(M = m)$)
  - But Bob can recover m by computing $x \oplus k$
- The main problem is that k has to be as long as m and cannot be reused so... how to agree on k?

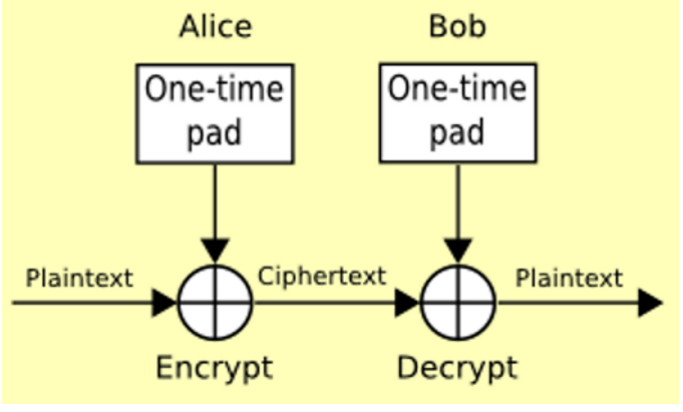


Image credits: nullprogram.com

The problem of key distribution

- Alice and Bob may share several keys for later use when they are together
- But... what if they cannot meet each other?
- There exist key distribution methods like the Diffie-Hellman protocol but...
  - They are not unconditionally secure (they usually rely on hardness assumptions)
  - In fact, DH can be broken with quantum computers!

BB84: Alice's part

- In 1984, Charles Bennett and Gilles Brassard proposed the first protocol for quantum key distribution (QKD)
- Alice generates a (private) string of random bits
- She could even do this with a quantum computer (*H* gate + measure)
- Then, for each bit she randomly chooses if she encodes it in the $\{|0\rangle, |1\rangle\}$ basis or in the $\{|+\rangle, |-\rangle\}$ basis
- She can easily do this by using *H* and *X* gates
- Alice sends the resulting qubits to Bob (through a quantum but not necessarily secure channel)

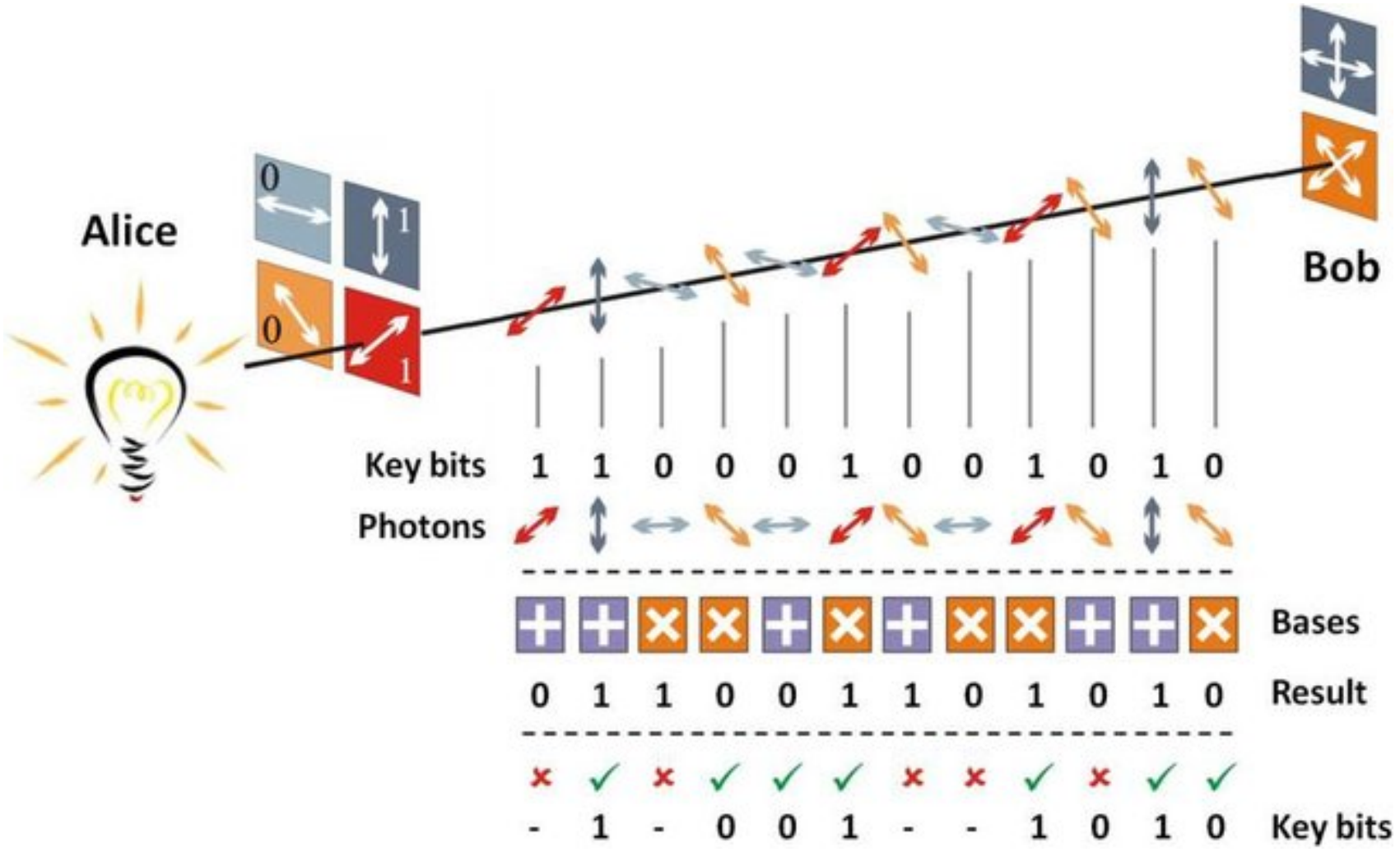
BB84: Bob's part

- Each time Bob receives a qubit, he randomly decides whether he will measure it in the $\{|0\rangle, |1\rangle\}$ basis or in the $\{|+\rangle, |-\rangle\}$ basis
- He does this by applying (or not) the *H* gate before measuring
- He writes down the results and the basis he used:
  - If he used $\{|0\rangle, |1\rangle\}$ he writes down 0 if he gets $|0\rangle$ and 1 if he gets $|1\rangle$
  - If he used $\{|+\rangle, |-\rangle\}$ he writes down 0 if he gets $|+\rangle$ and 1 if he gets $|-\rangle$

BB84: Alice and Bob on the phone

- After this process, Alice and Bob talk on a classical channel (authenticated but not necessarily secure)
- Bob announces the bases he has used for the measurements and Alice announces the bases she used to code the bits
- Bob does NOT announce the results of his measurements
- For those bits in which Bob measured with the same basis that Alice used for coding, he has got the bit that Alice intended to send
- The rest are discarded (they will keep about half of the bits)

BB84: The protocol in an image

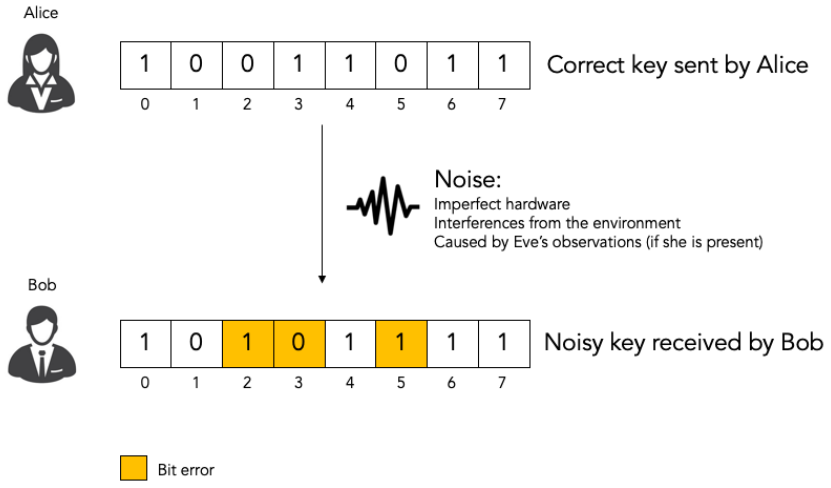


Eve tries to intercept and resend...

- Imagine Eve has access to the qubits that Alice sends to Bob
- Eve could try to measure and resend the qubit to Bob
- It is imposible for Eve to distinguish the four possibilities $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$ because she does not know the basis that Alice has chosen
- If Eve chooses a basis at random, she will make an error half of the time and Alice and Bob may detect it (by sharing some of the bits of the key to check that they are equal)
- Eve cannot copy the qubits and wait to check the basis that Alice and Bob have used (no cloning theorem)
- Other more complex attacks are possible, but can be shown to fail

Information reconciliation and privacy amplification

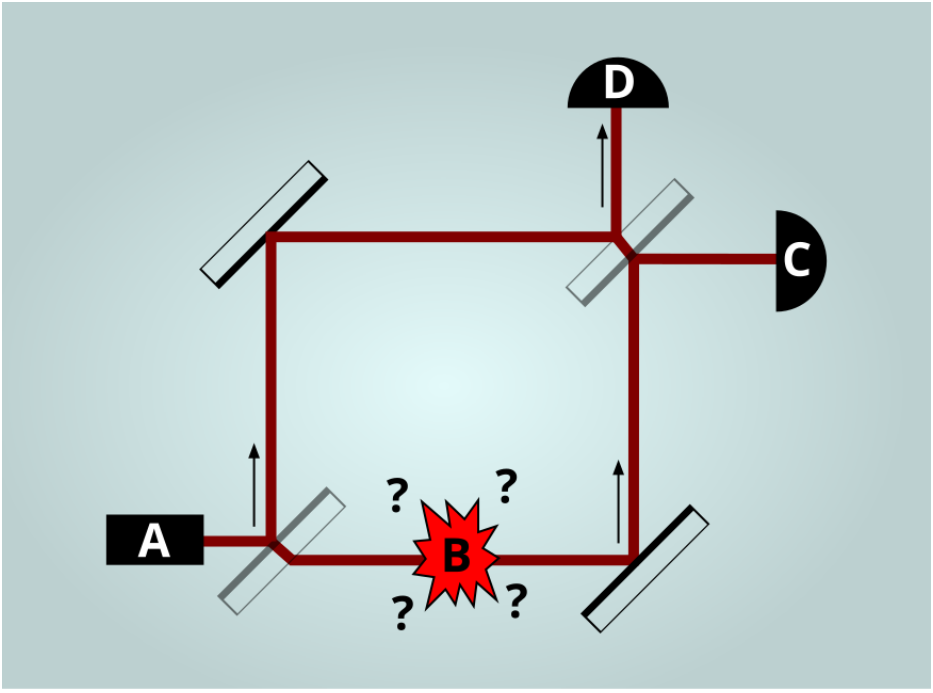
- Because of imperfections in the channel and devices or because of eavesdropping, some of the bits that Alice and Bob have may be different
- They can conduct a process of information reconciliation (for instance, with the cascade protocol)
- After this phase (or even before), some information may have leaked to Eve
- Alice and Bob can perform privacy amplification (for instance, with randomness extractors)



QKD at CERN

Other protocols that use independent qubits

- The use of independent qubits does not fully exploit the possibilities of quantum information, but there are some additional interesting applications
- For instance:
  - Other QKD protocols: B92, SARG04, Six-state protocol...
  - The concept of quantum money (Wiesner)
  - The Elitzur-Vaidman bomb tester



Part V

Two-qubit systems: more than the sum of their parts

Working with two qubits

- Each of the qubits can be in state $|0\rangle$ or in state $|1\rangle$
- So for two qubits we have four possibilities:

$$|0\rangle \otimes |0\rangle, |0\rangle \otimes |1\rangle, |1\rangle \otimes |0\rangle, |1\rangle \otimes |1\rangle$$

that we also denote

$$|0\rangle |0\rangle, |0\rangle |1\rangle, |1\rangle |0\rangle, |1\rangle |1\rangle$$

or

$$|00\rangle, |01\rangle, |10\rangle, |11\rangle$$

- Of course, we can have superpositions so a generic state is

$$|\psi\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle$$

where α_{xy} are complex numbers such that

$$\sum_{x,y=0}^{1} |\alpha_{xy}|^2 = 1$$

Measuring a two-qubit system

- Suppose we have a state

$$|\psi\rangle = \alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle + \alpha_{10}\,|10\rangle + \alpha_{11}\,|11\rangle$$

- If we measure both qubits, we will obtain:
  - 00 with probability $|\alpha_{00}|^2$ and the new state will be $|00\rangle$
  - 01 with probability $|\alpha_{01}|^2$ and the new state will be $|01\rangle$
  - 10 with probability $|\alpha_{10}|^2$ and the new state will be $|10\rangle$
  - 11 with probability $|\alpha_{11}|^2$ and the new state will be $|11\rangle$
- It is an analogous situation to what we had with one qubit, but now with four possibilities

Measuring just one qubit in a two-qubit system

- If we have a state

$$|\psi\rangle = \alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle + \alpha_{10}\,|10\rangle + \alpha_{11}\,|11\rangle$$

  we can also measure just one qubit
- If we measure the first qubit (for the second one is analogous):
  - We will get 0 with probability $|\alpha_{00}|^2 + |\alpha_{01}|^2$
  - In that case, the new state of $|\psi\rangle$ will be

$$\frac{\alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}}$$

  - We will get 1 with probability $|\alpha_{10}|^2 + |\alpha_{11}|^2$
  - In that case, the new state of $|\psi\rangle$ will be

$$\frac{\alpha_{10}\,|10\rangle + \alpha_{11}\,|11\rangle}{\sqrt{|\alpha_{10}|^2 + |\alpha_{11}|^2}}$$

Two-qubit states and vector representation

- A general two-qubit quantum state is

$$|\psi\rangle = \alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle + \alpha_{10}\,|10\rangle + \alpha_{11}\,|11\rangle$$

- We can represent with the column vector

$$\begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix}$$

- We can compute inner products by noticing that

$$\langle 00|00\rangle = \langle 01|01\rangle = \langle 10|10\rangle = \langle 11|11\rangle = 1$$

$$\langle 00|01\rangle = \langle 00|10\rangle = \langle 00|11\rangle = \cdots = \langle 11|00\rangle = 0$$

- A two-qubit quantum gate is a unitary matrix U of size 4×4

Tensor product of one-qubit gates

- The simplest way of obtaining a two-qubit gate is by having a pair of one-qubit gates A and B acting on each of the qubits
- In this case, the matrix for the two-qubit gate is the tensor product $A \otimes B$
- It holds that

$$(A \otimes B)(|\psi_1\rangle \otimes |\psi_2\rangle) = (A\,|\psi_1\rangle) \otimes (B\,|\psi_2\rangle)$$

- Of course, either A or B may be the identity
- This does NOT exhaust all posible two-qubit gates

$$\begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} \otimes \begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} = \begin{bmatrix} a_{1,1}\begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} & a_{1,2}\begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} \\ a_{2,1}\begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} & a_{2,2}\begin{bmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} a_{1,1}b_{1,1} & a_{1,1}b_{1,2} & a_{1,2}b_{1,1} & a_{1,2}b_{1,2} \\ a_{1,1}b_{2,1} & a_{1,1}b_{2,2} & a_{1,2}b_{2,1} & a_{1,2}b_{2,2} \\ a_{2,1}b_{1,1} & a_{2,1}b_{1,2} & a_{2,2}b_{1,1} & a_{2,2}b_{1,2} \\ a_{2,1}b_{2,1} & a_{2,1}b_{2,2} & a_{2,2}b_{2,1} & a_{2,2}b_{2,2} \end{bmatrix}$$

The *CNOT* gate

- The *CNOT* (or controlled-*NOT* or *cX*) gate is given by the (unitary) matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

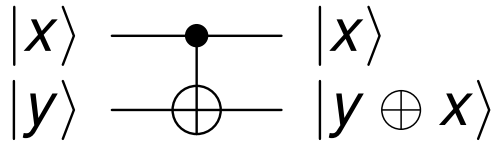
- If the first qubit is $|0\rangle$, nothing changes. If it is $|1\rangle$, we flip the second bit (and the first stays the same)
- That is:

$$|00\rangle \to |00\rangle \qquad |01\rangle \to |01\rangle$$

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

Action of the *CNOT* gate

- Its action on $x, y \in \{0, 1\}$ is, then:



- This is an extremely important gate for it allows to:
  - Create entanglement (more on this soon)
  - Copy *classical* information, because:

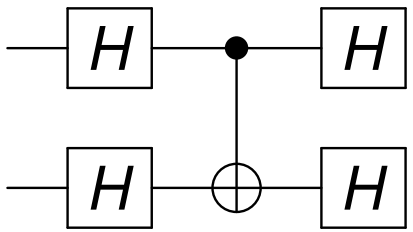
$$|00\rangle \rightarrow |00\rangle$$

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

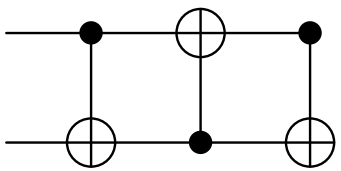
  - Construct other controlled gates

Equivalences with CNOT gates

- Sometimes, CNOT gates are not implemented between all pairs of qubits in a quantum computer
- We can use *H* gates to change the control and target of a CNOT gate



- We can swap states using three CNOT gates

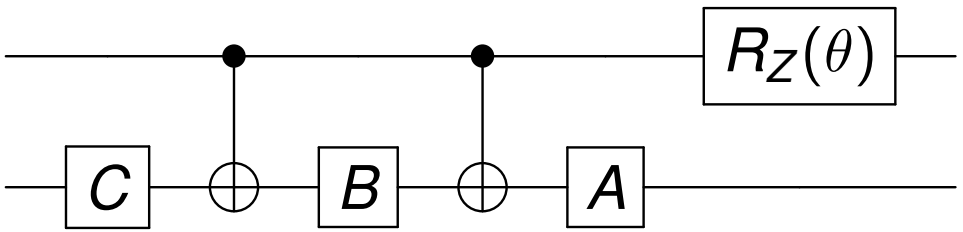


Constructing controlled gates by using the *CNOT* gate

- Any one-qubit gate U can be decomposed in the form

$$e^{i\theta} AXBXC$$

  with $ABC = I$
- Then, the circuit



  implements a U gate on the lower qubit controlled by the upper qubit

The no-cloning theorem

- There is **no** quantum gate that makes copies of an arbitrary (unknown) qubit
- The proof is easy: suppose we have a gate U such that $U|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle$
- Then $U|00\rangle = |00\rangle$ and $U|10\rangle = |11\rangle$ and by linearity

$$U(\frac{1}{\sqrt{2}}(|00\rangle+|10\rangle)) = \frac{1}{\sqrt{2}}(U|00\rangle+U|10\rangle) = \frac{1}{\sqrt{2}}(|00\rangle+|11\rangle)$$

- But

$$\frac{|00\rangle + |10\rangle}{\sqrt{2}} = (\frac{|0\rangle + |1\rangle}{\sqrt{2}})|0\rangle$$

so we should have

$$U(\frac{|00\rangle + |10\rangle}{\sqrt{2}}) = \frac{(|0\rangle + |1\rangle)}{\sqrt{2}}\frac{(|0\rangle + |1\rangle)}{\sqrt{2}} \neq \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Quantum entanglement: the spooky action at a distance

- We say that a state $|\psi\rangle$ is a product state if it can be written in the form

$$|\psi\rangle = |\psi_1\rangle\,|\psi_2\rangle$$

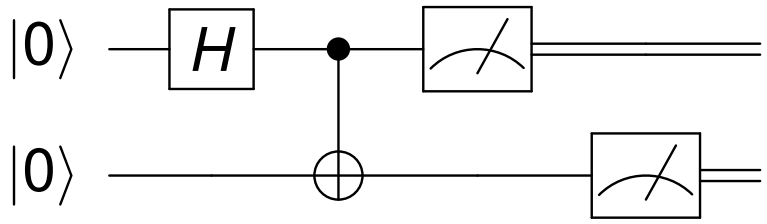
  where $|\psi_1\rangle$ and $|\psi_2\rangle$ are two states (of at least one qubit)
- An **entangled** state is a state that is not a product state
- Example of entangled states (Bell states):

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}} \qquad \frac{|00\rangle - |11\rangle}{\sqrt{2}}$$

$$\frac{|01\rangle + |10\rangle}{\sqrt{2}} \qquad \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$

Hello, entangled world!

- We can construct (and measure) Bell states with simple circuits



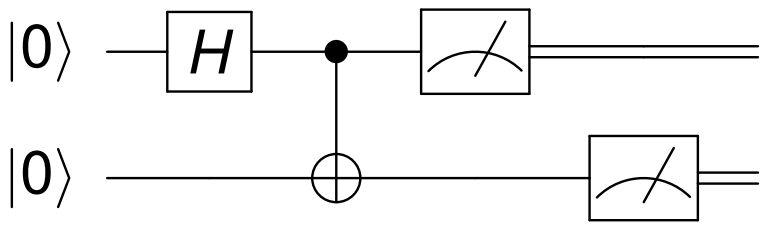
- Initially, the state of the system is $|00\rangle$
- After we apply the *H* gate, the state is

$$\frac{|00\rangle + |10\rangle}{\sqrt{2}}$$

- When we apply the *CNOT* gate, the state changes to

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

Hello, entangled world!

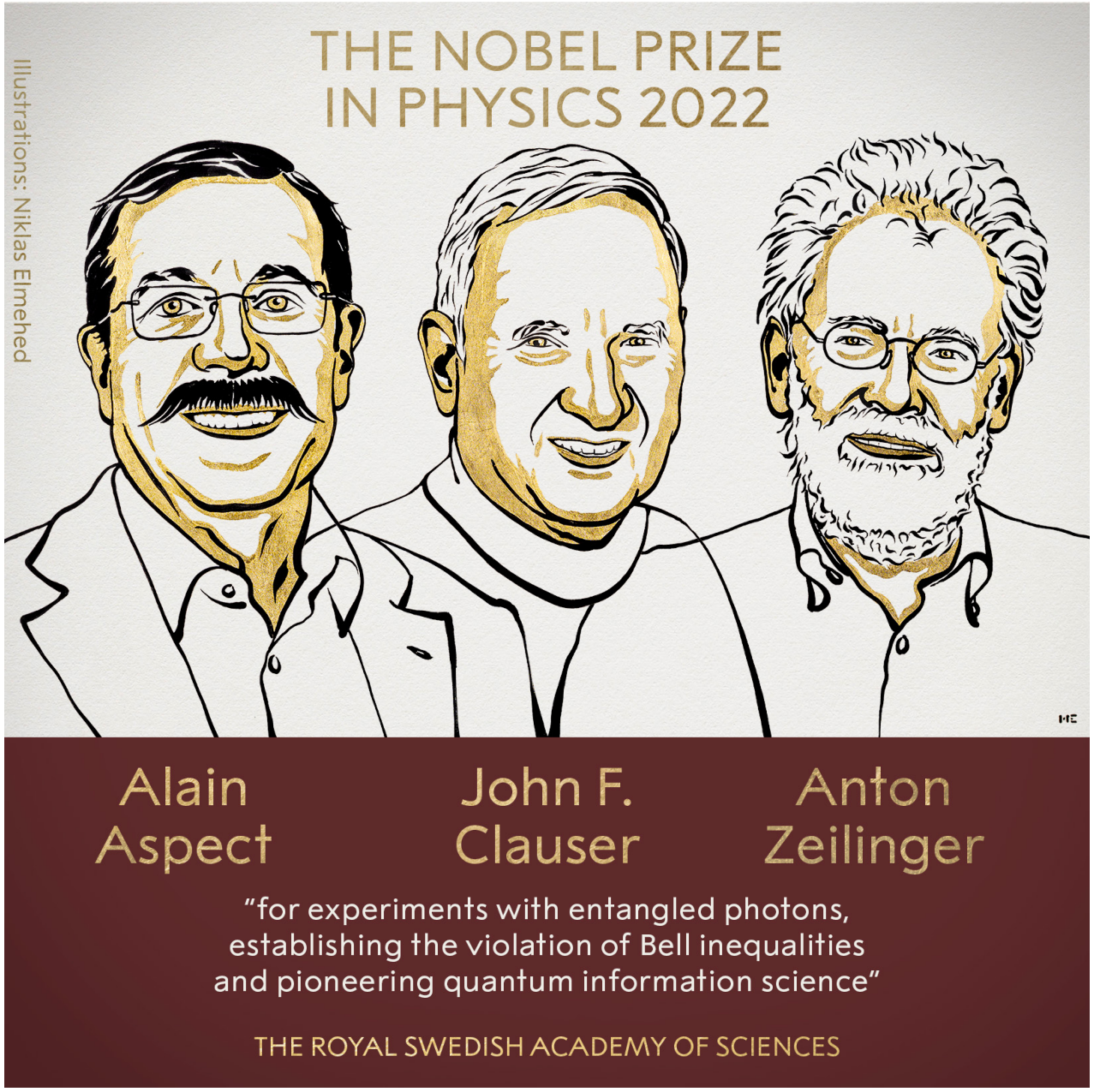


- Before we measure the first qubit, we have the state $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$
- We will get 0 or 1, each with probability $\frac{1}{2}$
- Suppose we obtain 0. Then, the new state will be $|00\rangle$
- Then, when we measure the second qubit we will obtain 0 with probability 1!
- Also, if we obtain 1 in the first qubit, in the second we will also obtain 1!

Part VI

The CHSH game: Nature isn't classical, dammit

Nobel Prize 2022

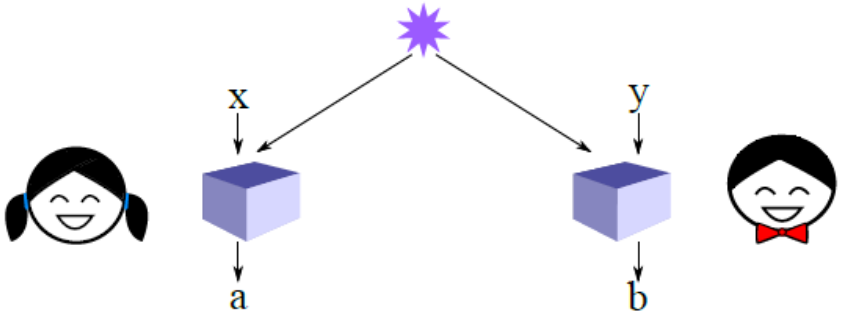


The CHSH game

- Based in an inequality proposed in 1969 by Clauser, Horne, Shimony and Holt based on previous work by John Bell
- Alice and Bob receive bits x and y from a referee
- They have to respond with bits a and b
- They win if

$$a \oplus b = x \cdot y$$

- They can decide on a joint strategy beforehand, but they cannot communicate during the game



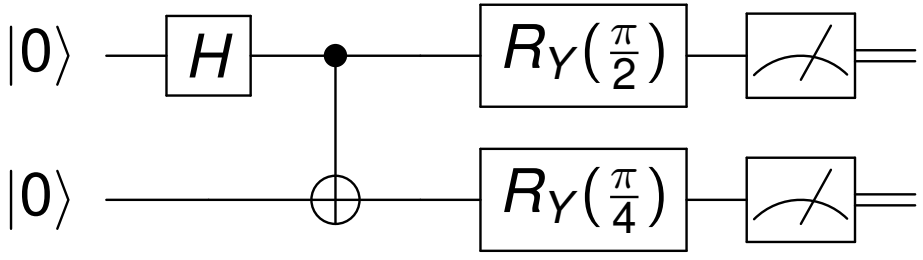
Classical strategies for the CHSH game

- Alice and Bob can win 75% of the time if they always answer ‘0’
- No other deterministic strategy can do better
- And probabilistic strategies are convex combinations of classical strategies so they cannot improve the 75% success rate

	$a = 0$	$a = 1$	$a = x$	$a = \neg x$
$b = 0$	$3/4$	$1/4$	$3/4$	$1/4$
$b = 1$	$1/4$	$3/4$	$1/4$	$3/4$
$b = y$	$3/4$	$1/4$	$1/4$	$3/4$
$b = \neg y$	$1/4$	$3/4$	$3/4$	$1/4$

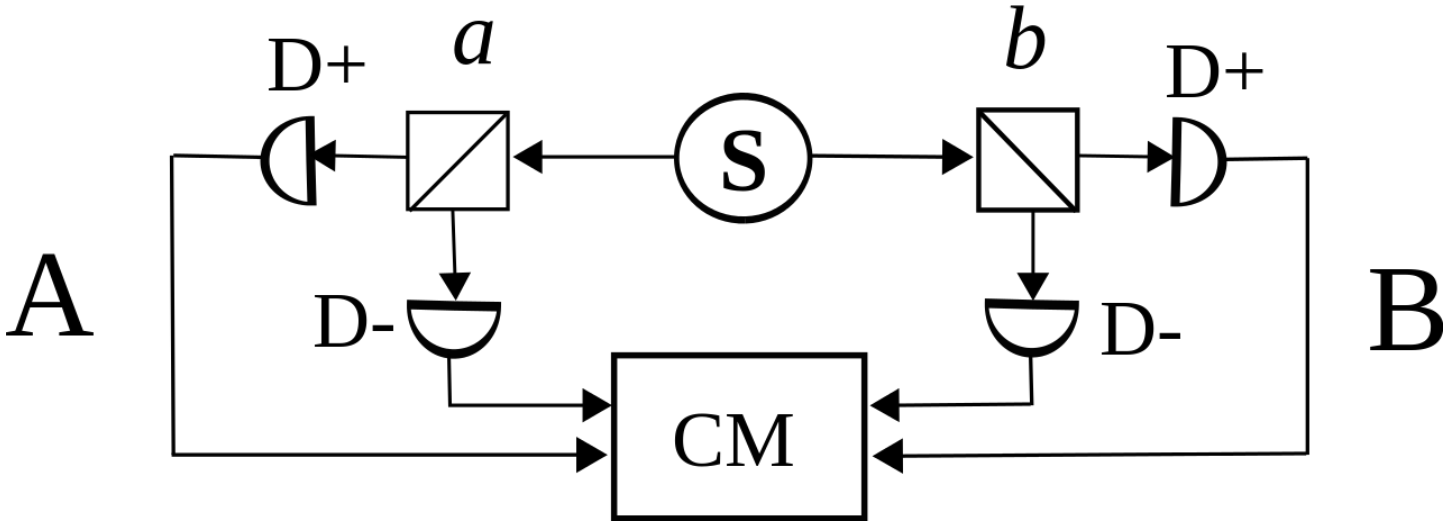
Quantum strategy for the CHSH game

- Alice and Bob share a Bell pair $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$ before the start of the game
- If Alice receives 0, she measures her qubit and ouputs the result
- If she receives 1, she applies $R_Y(\frac{\pi}{2})$ to her qubit and then she measures it
- If Bob receives 0, he applies $R_Y(\frac{\pi}{4})$. Else, he applies $R_Y(-\frac{\pi}{4})$.
- Then, he measures his qubit
- The probability of winning is now $\cos^2(\frac{\pi}{8}) \approx 0.85 > 0.75$



Some comments on the CHSH game

- It can be proved that $\cos^2(\frac{\pi}{8})$ is the highest possible success rate for a quantum strategy (Tsirelson's bound)
- The CHSH game can be used to rule out local realism
- Several experiments have been conducted, including:
  - Aspect et al. (1981-82)
  - Hensen et al. (2005) - Eliminate the locality and detection loopholes
- All of them agree with the predictions of quantum theory



The GHZ game

- Introduced by Greenberger, Horne and Zeilinger
- A referee selects *rst* from $\{000, 011, 101, 110\}$ and sends r to Alice, s to Bob and t to *Charlie*
- They produce a, b and c and win if

$$a \oplus b \oplus c = r \vee s \vee t$$

- Classically, they can only win with 75% probability
- Quantumly, they can win every single time
  - They share the state

$$\frac{1}{2}(|000\rangle - |011\rangle - |101\rangle - |110\rangle)$$

  - They apply H to their qubit if the receive 1
  - They measure and return the answer
- This is sometimes called “quantum pseudo-telepathy” (Brassard, Cleve, Tapp)
- Both the CHSH and the GHZ game can be used for randomness certification (and expansion)

Part VII

Superdense coding: two for the price of one

Superdense coding: two for the price of one (more or less)

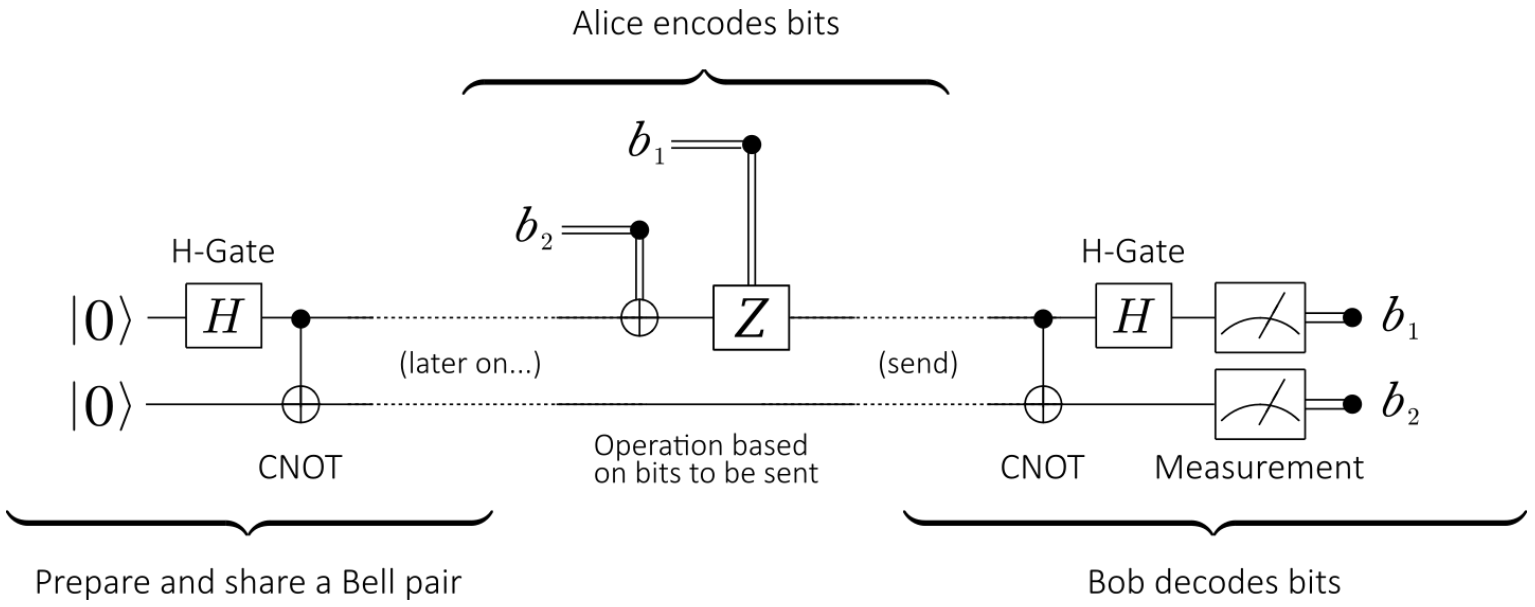
- How many classical bits can we communicate with one qubit?
- Holevo's bound: the accesible information of one qubit is just one bit
- However, if Alice and Bob share in advance a Bell pair... we can send two bits of information with just one qubit!

$$1 qubit + 1 ebit \geq 2 bits$$

- This protocol is, in some sense, the inverse of quantum teleportation

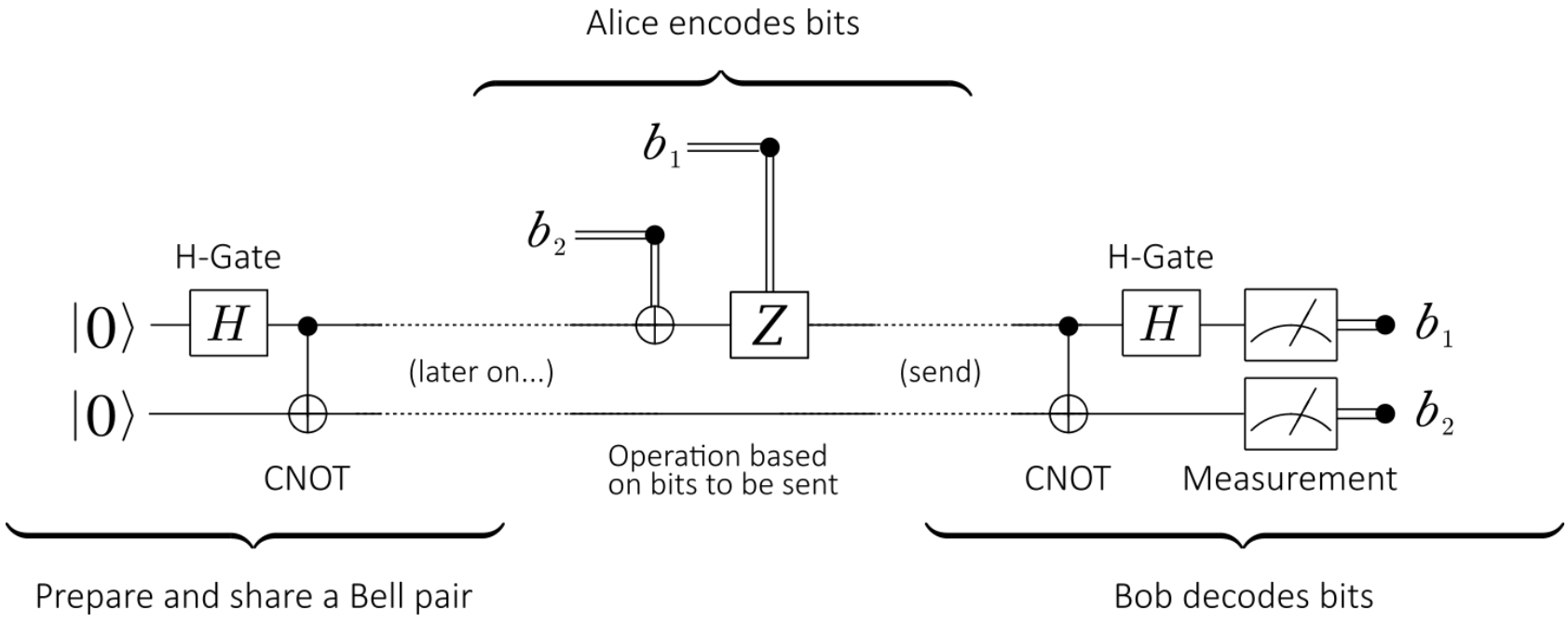
Superdense coding: Alice's part

- Alice and Bob share a Bell pair in advance $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$
- Alice wants to send to Bob two classical bits b_1 and b_2
- If $b_2 = 1$, she applies X to her qubit
- If $b_1 = 1$, she applies Z to her qubit
- Then, she sends her qubit to Bob



Superdense coding: Bob's part

- Bob receives Alice's qubit
- He applies a *CNOT* gate controlled by Alice's qubit
- He applies *H* to Alice's qubit
- He measures and recovers b_1 and b_2



Superdense coding: an example

- Suppose Alice wants to send 11
- We start with $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$
- After Alice's operations, we will have $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$
- When Bob applies *CNOT* he obtains

$$\frac{1}{\sqrt{2}}(|01\rangle - |11\rangle) = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\,|1\rangle$$

- And with the *H* gate he gets $|11\rangle$ that now he can measure

Part VIII

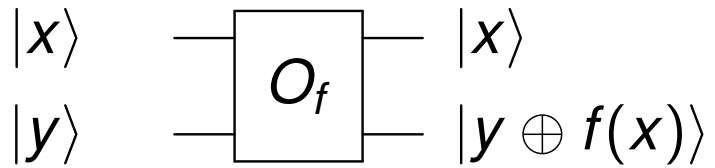
Deutsch's algorithm: the grandfather of all quantum algorithms

Deutsch's algorithm: statement of the problem

- In 1985, David Deutsch proposed a very simple algorithm that, nevertheless, hints at the capabilities of quantum computing
- The problem it solves is only of theoretical relevance and was later generalized in a joint work with Jozsa
- We are given a circuit (an **oracle**) that implements a one-bit boolean function and we are asked to determine whether the function is constant (returns the same value for all inputs) or balanced (returns 1 on one input and 0 on the other)
- Alternatively, we can think of the oracle as indexing a bit string of length two and we are asked to compute the XOR of the bits of the string
- In the classical case, we would need to consult the oracle twice, to compute both values of the function
- In the quantum case, we can make just one oracle call... but in superposition

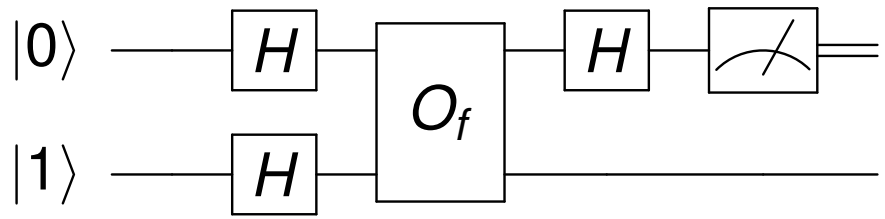
Deutsch's algorithm: the oracle

- An oracle is treated as a black box, a circuit whose interior we cannot know
- This circuit computes, in a reversible way, a certain function f (in our case, of just one input)
- For the computation to be reversible, it uses as many inputs as outputs and "writes the result" with an XOR



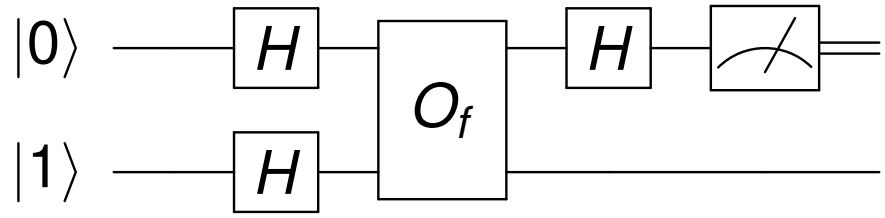
Deutsch's algorithm: the circuit

- The quantum circuit that we need to use to solve the problem is very simple



- If the function is constant, we will measure 0
- If the function is balanced, we will measure 1

Deutsch's algorithm: the magic



- The initial state is $|0\rangle\,|1\rangle$
- After the H the gates we have

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

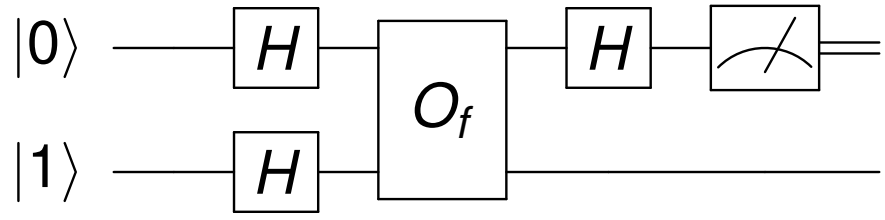
which is the same as

$$\frac{|0\rangle\,(|0\rangle - |1\rangle)}{2} + \frac{|1\rangle\,(|0\rangle - |1\rangle)}{2}$$

- When we apply the oracle, by linearity we obtain

$$\frac{|0\rangle\,(|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle)}{2} + \frac{|1\rangle\,(|0 \oplus f(1)\rangle - |1 \oplus f(1)\rangle)}{2}$$

Deutsch's algorithm: the magic (2)



- If $f(0) = 0$, we have

$$|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle = |0\rangle - |1\rangle$$

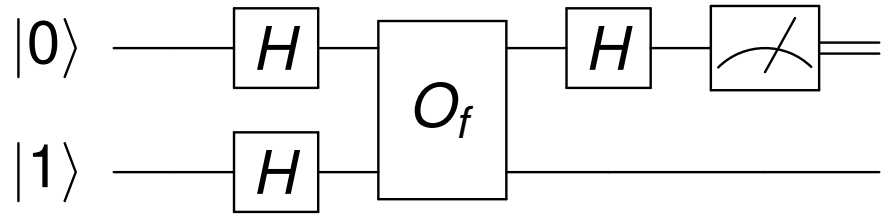
- However, if $f(0) = 1$ we get

$$|0 + f(0)\rangle - |1 \oplus f(0)\rangle = |0 \oplus 1\rangle - |1 \oplus 1\rangle = |1\rangle - |0\rangle = -(|0\rangle - |1\rangle)$$

- For $f(1)$ the situation is the same so the global state is

$$\frac{(-1)^{f(0)}\,|0\rangle\,(|0\rangle - |1\rangle)}{2} + \frac{(-1)^{f(1)}\,|1\rangle\,(|0\rangle - |1\rangle)}{2}$$

Deutsch's algorithm: the magic (3)



- We can also write that state as

$$\frac{|0\rangle\,(|0\rangle - |1\rangle)}{2} + \frac{(-1)^{f(0)+f(1)}\,|1\rangle\,(|0\rangle - |1\rangle)}{2}$$

- So if $f(0) = f(1)$, we will have

$$\frac{|0\rangle\,(|0\rangle - |1\rangle)}{2} + \frac{|1\rangle\,(|0\rangle - |1\rangle)}{2} = \frac{(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)}{2}$$

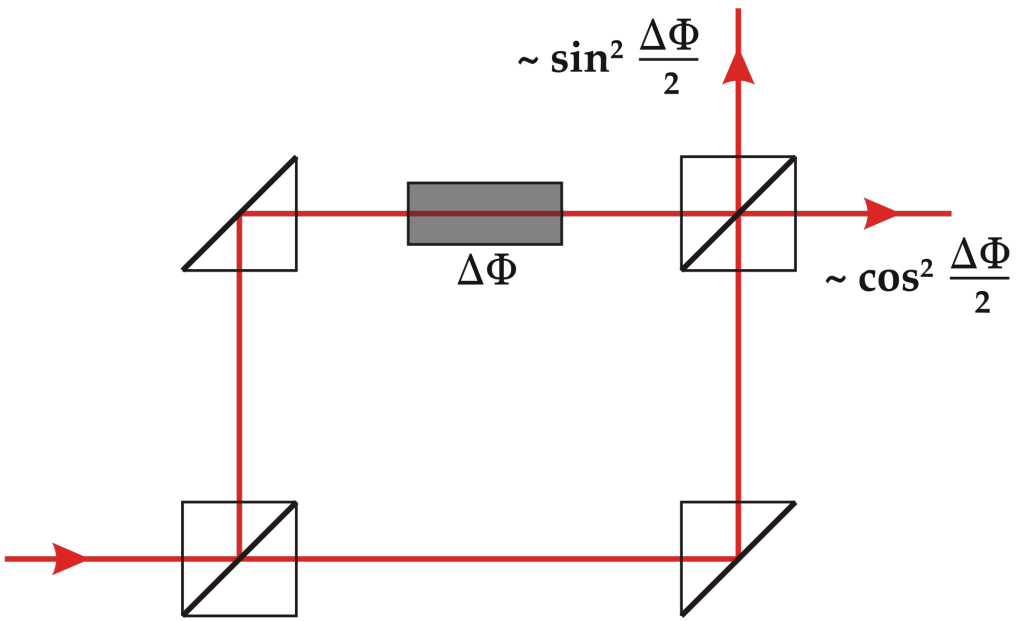
  and when we apply the last H and measure we obtain 0.
- But if $f(0) \neq f(1)$, the state is

$$\frac{|0\rangle\,(|0\rangle - |1\rangle)}{2} - \frac{|1\rangle\,(|0\rangle - |1\rangle)}{2} = \frac{(|0\rangle - |1\rangle)(|0\rangle - |1\rangle)}{2}$$

  and, then, we obtain 1.

Deutsch's algorithm: some comments

- When we apply the oracle we have a phase kickback: we only act on one qubit, but it affects the whole state
- Deutch's algorithm exploits an interference phenomenon similar to that found in some physical experiments (double-slit experiment, Mach-Zehnder interferometer)



Part IX

Multiqubit systems: growing up!

n-qubit systems

- When he have n qubits, each of them can be in state $|0\rangle$ and $|1\rangle$
- Thus, for the n-qubit state we have 2^n possibilities:

$$|00\ldots0\rangle, |00\ldots1\rangle, \ldots, |11\ldots1\rangle$$

or simply

$$|0\rangle, |1\rangle, \ldots, \left|2^n-1\right\rangle$$

- A generic state of the system will be

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \ldots + \alpha_{2^n-1} \left|2^n-1\right\rangle$$

where α_i are complex numbers such that

$$\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$$

Measuring a *n*-qubit state

- Suppose we have the *n*-qubit state

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \ldots + \alpha_{2^n-1} \left|2^n - 1\right\rangle$$

- If we measure all its qubits, we obtain:
  - 0 with probability $|\alpha_0|^2$ and the new state will be $|0 \ldots 00\rangle$
  - 1 with probability $|\alpha_1|^2$ and the new state will be $|0 \ldots 01\rangle$
  - $\ldots$
  - $2^n - 1$ with probability $|\alpha_{2^n-1}|^2$ and the new state will be $|1 \ldots 11\rangle$
- It is analogous to what we had with one and two qubits, but now with 2^n possibilities

Measuring one qubit in a *n*-qubit state

- We have

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \ldots + \alpha_{2^n-1} \left|2^n - 1\right\rangle$$

- If we measure the *j*-th qubit
  - We will get 0 with probability

$$\sum_{i \in I_0} |\alpha_i|^2$$

    where I_0 is the set of numbers whose *j*-th bit is 0
  - In that case, the new state $|\psi\rangle$ will be

$$\frac{\sum_{i \in I_0} \alpha_i |i\rangle}{\sqrt{\sum_{i \in I_0} |\alpha_i|^2}}$$

  - The case in which we obtain 1 is analogous

n-qubit quantum gates

- A *n*-qubit state is

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \ldots + \alpha_{2^n-1} \left|2^n - 1\right\rangle$$

- We can represent it by the column vector

$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2^n-1} \end{pmatrix}$$

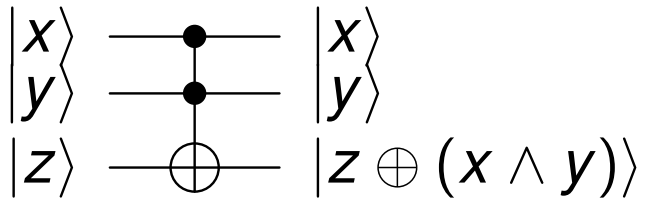
- To compute inner products with Dirac notation we only need to note that

$$\langle i|j\rangle = \delta_{ij}$$

- Thus, a *n*-qubit quantum gate is a unitary matrix U of size $2^n \times 2^n$

The Toffoli gate

- The Toffoli gate (or *CCNOT*) is a 3-qubit gate. Thus, it can be represented as a 8×8 matrix
- Its action on elements $x, y, z \in \{0, 1\}$ is:



- The Toffoli gate is **universal for classical logic**, and thus **any classical circuit can be simulated with a quantum circuit**
- However, the Toffoli gate, on its own, **is not universal for quantum computing** (and it is not even necessary, because it can be simulated with one and two-qubit gates)

Universal gates in quantum computing

- The number of quantum gates (even for a single qubit) is uncountably infinite. Thus, no finite set of gates is universal in the classical sense
- However, we can obtain finite sets of gates that allow us to **approximate** any other gate as much as we want

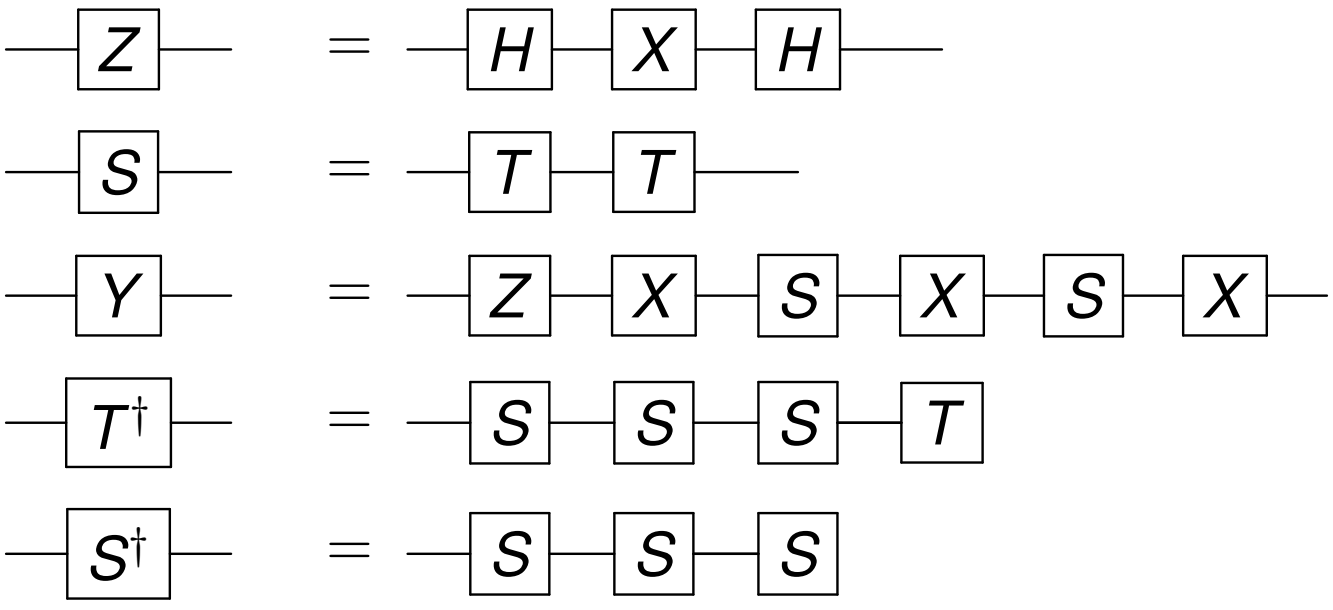
Theorem

The one-qubit gates together with the CNOT gate are (exactly) universal for quantum computing

Theorem

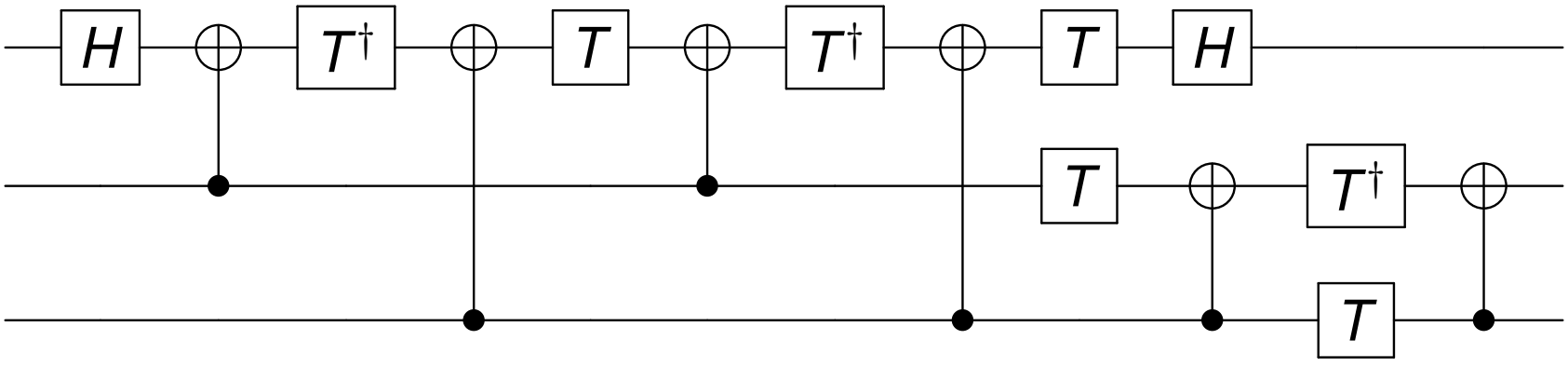
The gates X, H, T and CNOT are universal for quantum computing

Gate equivalences



However, Z, S, Y, $S^\dagger$ and $T^\dagger$ are usually included among the available gates in most quantum computers (such as the ones in IBM Quantum).

Equivalence of the Toffoli gate

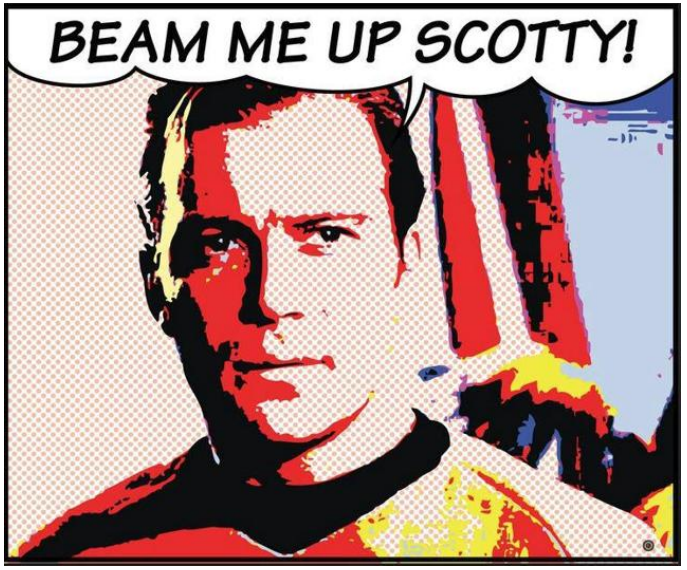


Part X

Quantum teleportation: to boldly transmit quantum states

Quantum teleportation: Quantum me up, Scotty!

- Can Alice sent a qubit $|\psi\rangle$ to Bob it there is no quantum channel available?
- We are interested in the most general case, even if Alice does not know which state she has
- The problem can be solved if Alice and Bob share an entangled state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$



Quantum teleportation: Alice's part

- Alice and Bob share an entangled state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$
  - This can be done in advance
  - Or they can rely on a source that distributes entangled pairs
- Alice applies a CNOT gate to the qubit she wants to teleport $|\psi\rangle = a|0\rangle + b|1\rangle$ and to her part of the Bell pair. We will have

$$\frac{1}{\sqrt{2}}(a(|000\rangle + |011\rangle) + b(|110\rangle + |101\rangle))$$

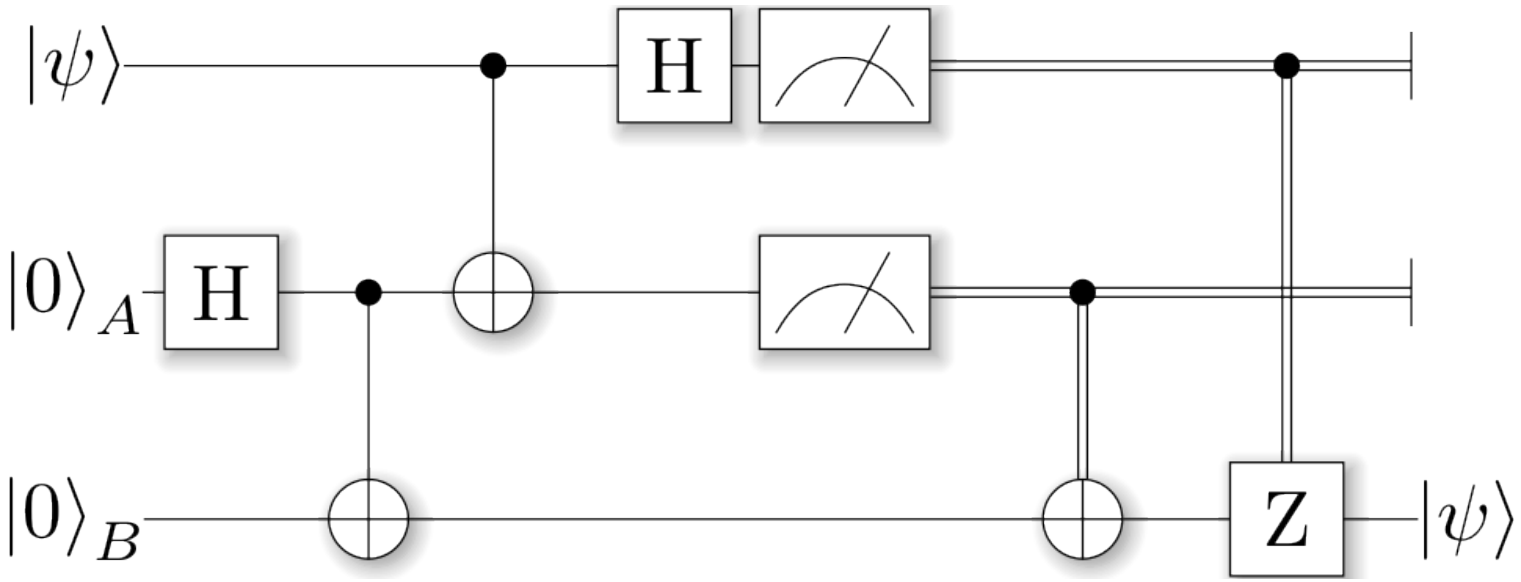
- Alice further applies the H gate to the qubit she wants teleported. Then, we have

$$\frac{1}{2}\big(|00\rangle(a|0\rangle + b|1\rangle) + |01\rangle(b|0\rangle + a|1\rangle) + |10\rangle(a|0\rangle - b|1\rangle) + |11\rangle(-b|0\rangle + a|1\rangle)\big)$$

- Alice measures her two qubits and sends the result (two classical bits) to Bob (through a classical channel)

Quantum teleportation: Bob's part

- Bob uses the second bit received from Alice to decide if he applies X to his qubit
- And he uses the first bit to decide if he applies Z



Quantum teleportation: some comments

- It is not matter that is teleported but information
- When Alice measure her qubit, she looses it (if not, we would be contradicting the no-cloning theorem)
- To teleport a qubit, we need two classical bits and one entangled pair:

$$2 bits + 1 ebit \geq 1 qubit$$

- Teleportation is not instantaneous, we need classical communication (no-communication theorem)
- Quantum teleportation has been shown experimentally (current record is 1,400 km)
- Demonstration of quantum teleportation in Quirk

Entanglement swapping

- Quantum teleportation can also be used with entangled qubits
- Alice shares a Bell pair with Bob and another one with Charlie
- In the figure, the top and bottom qubits belong to Alice. The second from the top belongs to Bob and the other to Charlie
- Alice teleports her top qubit to Charlie
- Now Bob's and Charlie's qubits are entangled (although maybe they were never in direct contact)

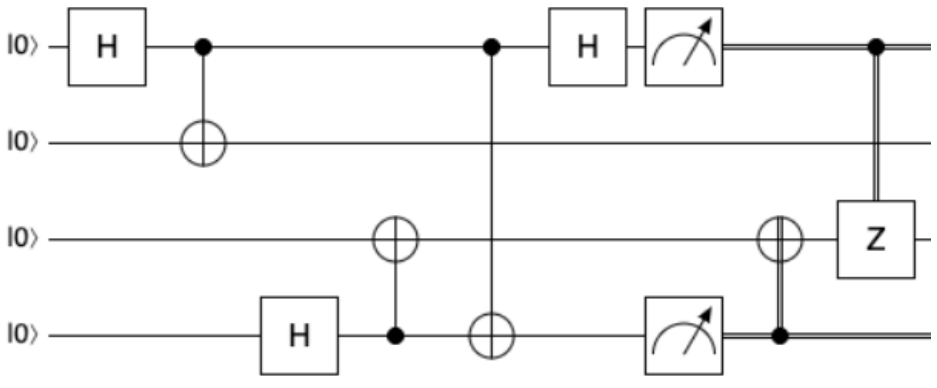
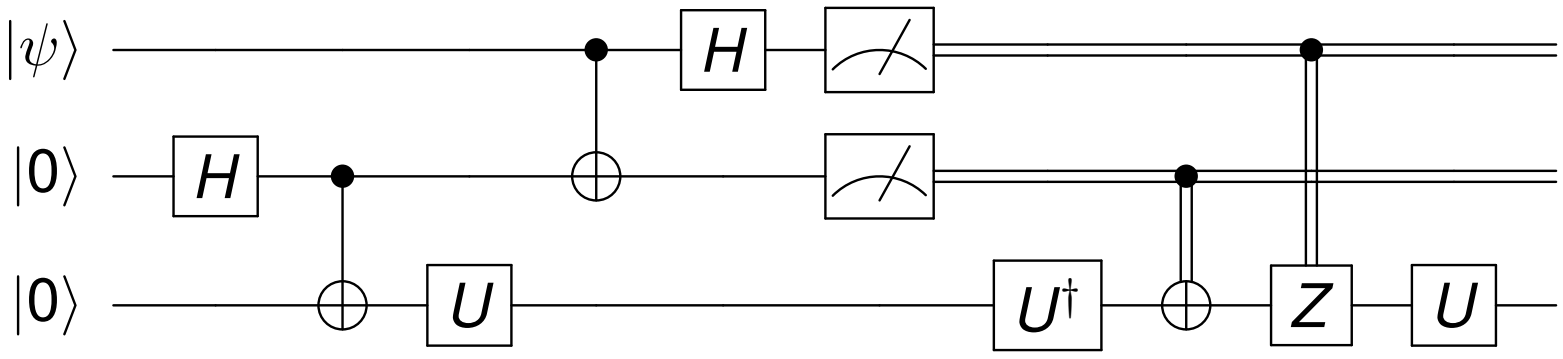


Image credits: Created with Quirk. Click here to access the circuit

Gate teleportation

- We can generalize the idea of quantum teleportation to teleport the action of gates
- With the circuit of the figure, we can apply gate U to an arbitrary state $|\psi\rangle$
- This is useful if preparing $\frac{1}{\sqrt{2}}(|0\rangle\, U\, |0\rangle + |1\rangle\, U\, |1\rangle)$ and applying $UXU^\dagger$, $UZU^\dagger$, $UZXU^\dagger$ are easy compared to applying U to a general qubit
- Such a situation can happen when $U = T$ in the context of fault-tolerant quantum computing



Part XI

Everything you always wanted to know about quantum parallelism but were afraid to ask

Urban legends about quantum parallelism

- But... don't quantum computers try all 2^n possibilities in parallel?
- The answer is... yes *and* no (this is *quantum* computing after all!)

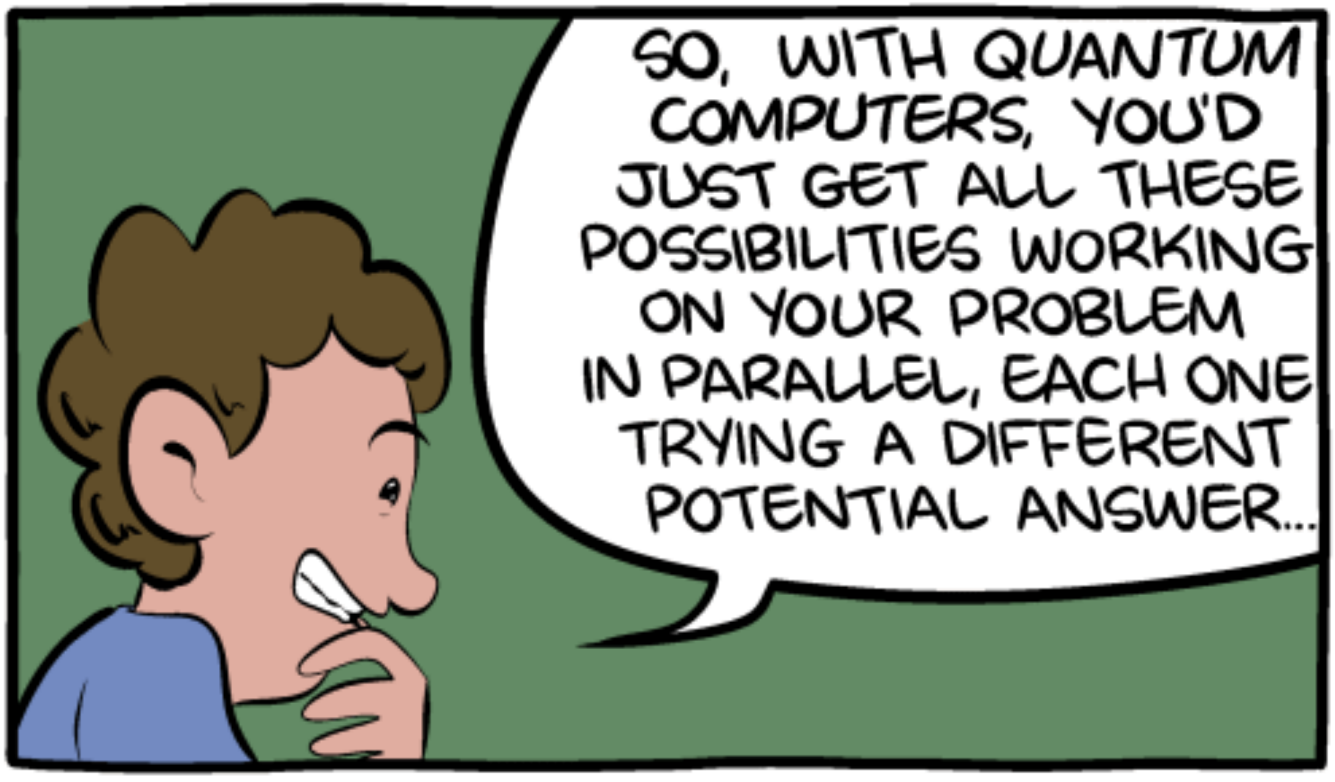
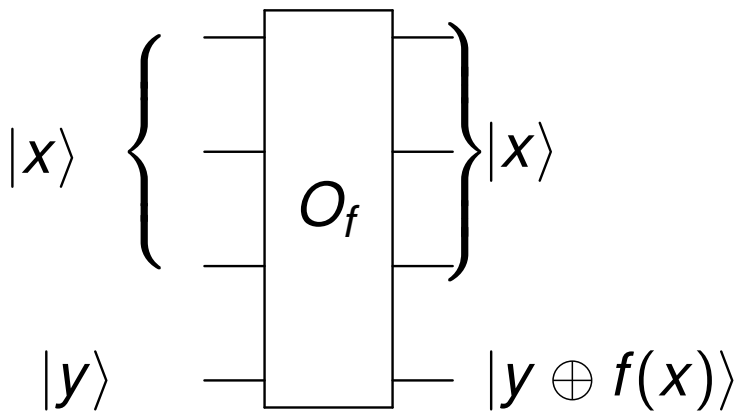


Image credits: The Talk, by Scott Aaronson and Zach Weinersmith

Evaluating a function: querying the oracle

- As we know, in quantum computing every gate is reversible
- To compute a function f we keep the inputs unchanged and *xor* the result to the output qubits
- This type of circuit is called and oracle for f (we have already used an oracle for a one-bit function in Deutsch's algorithm)



Evaluating a function in parallel: the superposition hocus-pocus

- Suppose that we have an oracle O_f for a function $f(x)$ with a one-bit input
- We know that, using the H gate, we can put a qubit in superposition
- If we start with the state $|0\rangle\,|0\rangle$ and we apply H on the first qubit, we will have

$$\frac{1}{\sqrt{2}}\,|0\rangle\,|0\rangle + \frac{1}{\sqrt{2}}\,|1\rangle\,|0\rangle$$

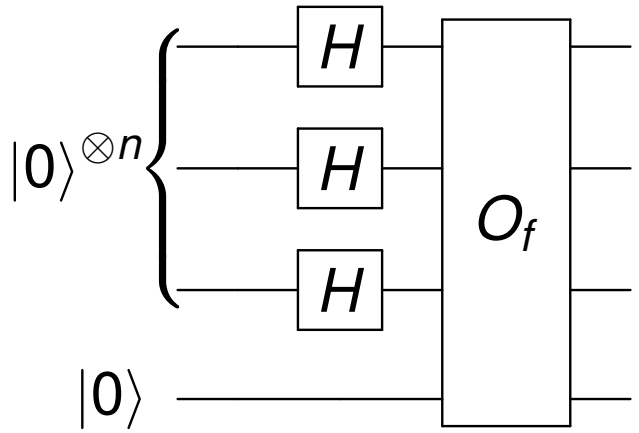
- If we now apply O_f, by linearity we have

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

- We have evaluated the function on two different inputs with just one call!

Evaluating a function in parallel: the tensor-product abracadabra

- We can do something similar with a function $f(x_1, x_2, \ldots, x_n)$ on n-variables by using the following circuit



- When we apply the H gates we obtain

$$\frac{(|0\rangle + |1\rangle)(|0\rangle + |1\rangle)\cdots(|0\rangle + |1\rangle)\,|0\rangle}{\sqrt{2^n}}$$

Evaluating a function in parallel: the tensor-product abracadabra (2)

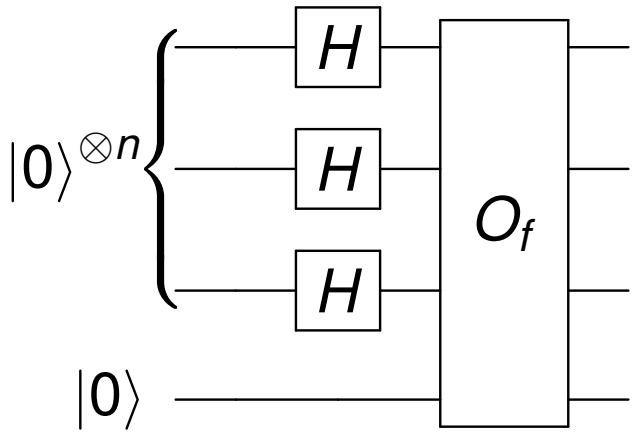
- If we expand the product we get

$$\frac{(|0\ldots0\rangle + |0\ldots1\rangle + \ldots + |1\ldots1\rangle)\,|0\rangle}{\sqrt{2^n}} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle\,|0\rangle$$

- And, when we apply the oracle, we will get the state

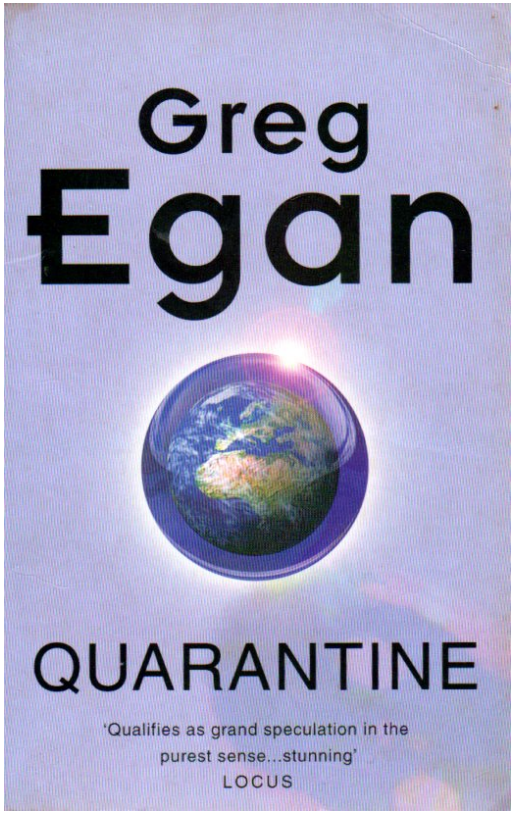
$$\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle\,|f(x)\rangle$$

- An exponential number of function evaluations with just one call!



Quantum parallelism vs. non-deterministic machines

- With a non-deterministic machine, we could choose at will some value *f*
- This would allow us to solve *NP*-complete problems
- A similar idea is used in the plot of *Quarantine*, a science-fiction novel by Greg Egan



All that glitters ain't gold

- And now... how do we retrieve the values $f(x)$?
- To obtain a result, we need to perform a measurement
- But then we will get a state of the form

$$|c\rangle\,|f(c)\rangle$$

- That is, we only obtain the result of the function for a randomly chosen input (this may be even worse than classically evaluating the function)

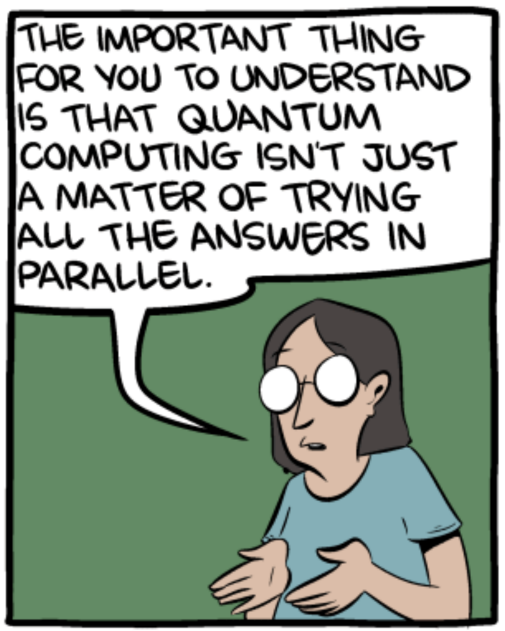


Image credits: The Talk, by Scott Aaronson and Zach Weinersmith

Interferences come to the rescue

- How can we use the 2^n evaluations to extract useful information?
- One possibility is... to produce interferences!
- The amplitudes of some states can be negative
- If we manage to “annihilitate” the amplitudes of states we are not interested in, the probability of obtaining the answer that we need will grow
- This is, in general, no easy task, but we know how to achieve it in some interesting cases

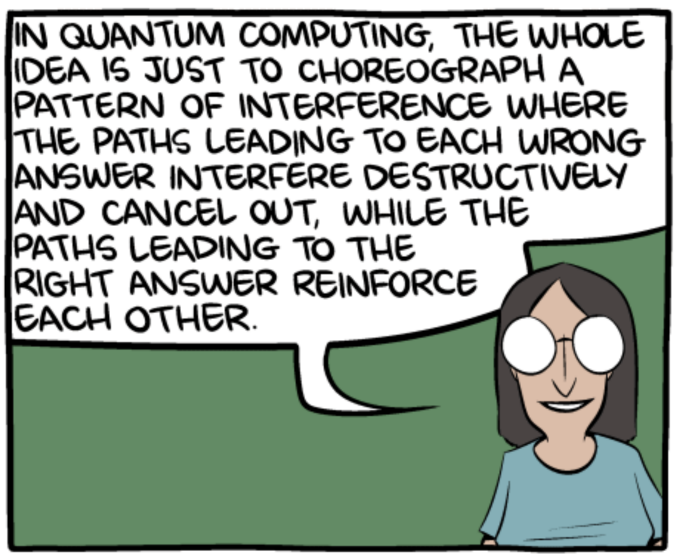


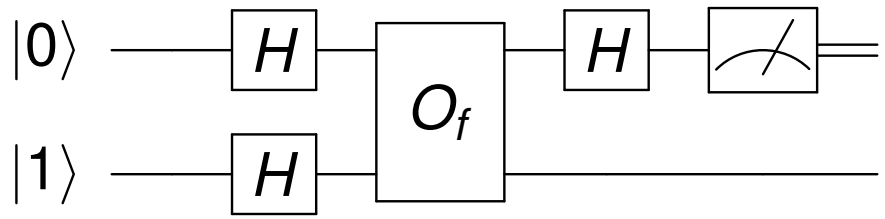
Image credits: The Talk, by Scott Aaronson and Zach Weinersmith

Part XII

The Deutsch-Jozsa algorithm: a very fast way of solving a problem that nobody asked to solve

Reminder: Deutsch's algorithm

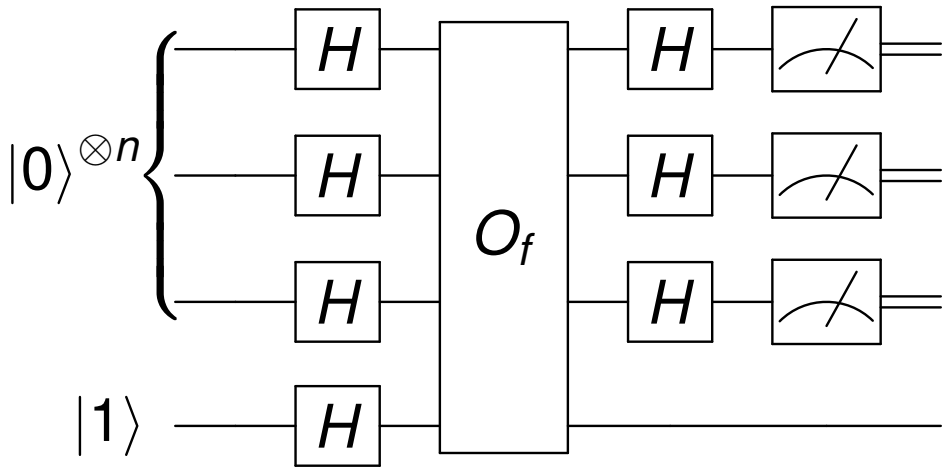
- We have an oracle O_f for a boolean function $f(x)$
- f can be constant (returns the same value for all inputs) or balanced (returns 1 on one input and 0 on the other)
- Distinguishing one situation from the other requires, in the classical case, evaluating the function on the two possible inputs
- With a quantum computer, we can solve the problem with just one call to O_f
- The key is to use quantum parallelism together with interference



Upping the ante: the Deutsch-Jozsa algorithm

- The Deutsch-Jozsa algorithm solves a type of problem called **promise problem**
  - We are given a boolean function $f(x_1, \ldots, x_n)$
  - We are **promised** that f is either constant (always 0 or 1) or balanced (0 for half of the inputs and 1 for the rest)
  - We have to decide which of the two cases we are in by calling the function as few times as possible
- With a classical deterministic algorithm we need (in the worst case) $2^{n-1} + 1$ calls to f
- With the Deutsch-Jozsa quantum algorithm it is enough to evaluate f **just once**

Circuit for the Deutsch-Jozsa algorithm



Steps in the Deutsch-Jozsa algorithm

1. We create the state $|0\ldots0\rangle\,|1\rangle$
2. We use Hadamard gates to create the superposition

$$\sum_{x\in\{0,1\}^n} \frac{1}{\sqrt{2^{n+1}}}\,|x\rangle\,(|0\rangle - |1\rangle)$$

3. We apply the oracle, getting

$$\sum_{x\in\{0,1\}^n} \frac{1}{\sqrt{2^{n+1}}}\,|x\rangle\,(|0\oplus f(x)\rangle - |1\oplus f(x)\rangle) =$$

$$\sum_{x\in\{0,1\}^n} \frac{(-1)^{f(x)}}{\sqrt{2^{n+1}}}\,|x\rangle\,(|0\rangle - |1\rangle)$$

Steps in the Deutsch-Jozsa algorithm (2)

4. We apply again Hadamard gates to the n first qubits and we obtain

$$\sum_{y\in\{0,1\}^n}\sum_{x\in\{0,1\}^n}\frac{(-1)^{f(x)+x\cdot y}}{2^n\sqrt{2}}\,|y\rangle\,(|0\rangle-|1\rangle)$$

5. Finally, we measure the n first qubits.
6. If the function is constant, we will obtain $|0\rangle$. Otherwise (if the function is balanced), we will get a string different from $|0\rangle$.

Correctness of the algorithm

- The probability of measuring $|0\rangle$ is exactly

$$\left(\sum_{x \in \{0,1\}^n} \frac{(-1)^{f(x)+x\cdot 0}}{2^n} \right)^2 = \left(\sum_{x \in \{0,1\}^n} \frac{(-1)^{f(x)}}{2^n} \right)^2$$

- If f is constant, the sum is 1
- If f is balanced, the sum is 0

Some comments on the Deutsch-Jozsa algorithm

- The problem we have solved is academical, with no practical interest
- But... it shows how quantum computing can obtain global information about a function with just one evaluation
- The key is to use:
  - Quantum parallelism (because of superposition)
  - Interference (constructive and destructive)
- Similar ideas are used in other algorithms, like the Bernstein-Vazirani and Simon methods

Part XIII

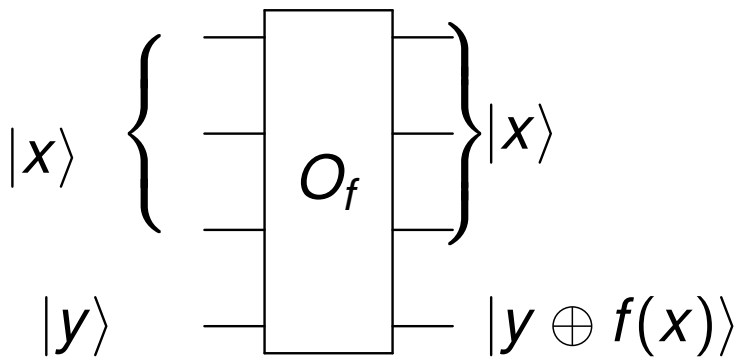
Grover's algorithm: finding the needle in the haystack

Statement of the problem

- Grover's algorithm is used to solve search problems
- Imagine we have an unsorted list of N elements
- One of them verifies a certain condition and we want to find it
- Any classical algorithm requires $O(N)$ queries to the list in the worst case
- Grover's algorithm can find the element with $O(\sqrt{N})$ queries

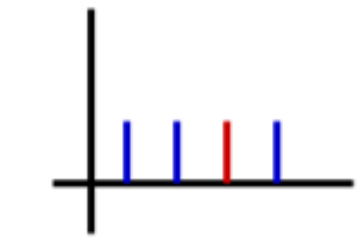
The oracle

- As in Deutsch-Jozsa's algorithm, we will use an oracle
- This oracle computes the function $f : \{0,1\}^n \Rightarrow \{0,1\}$ (with $N = 2^n$)
- The element we want to find is the one that verifies $f(x) = 1$

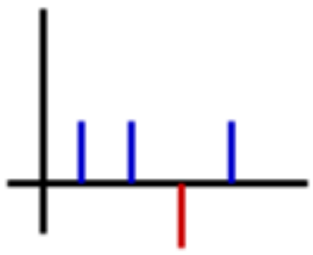


The idea behind the algorithm

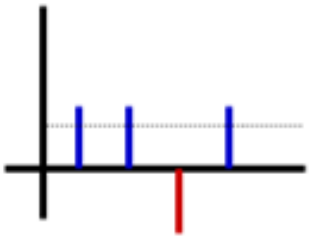
- Grover's algorithm is based on the idea of **inversion about the mean**



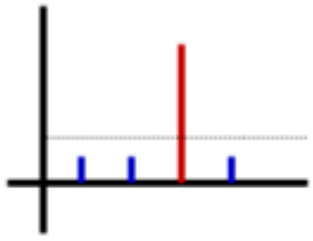
Original Amplitudes



Negate Amplitude



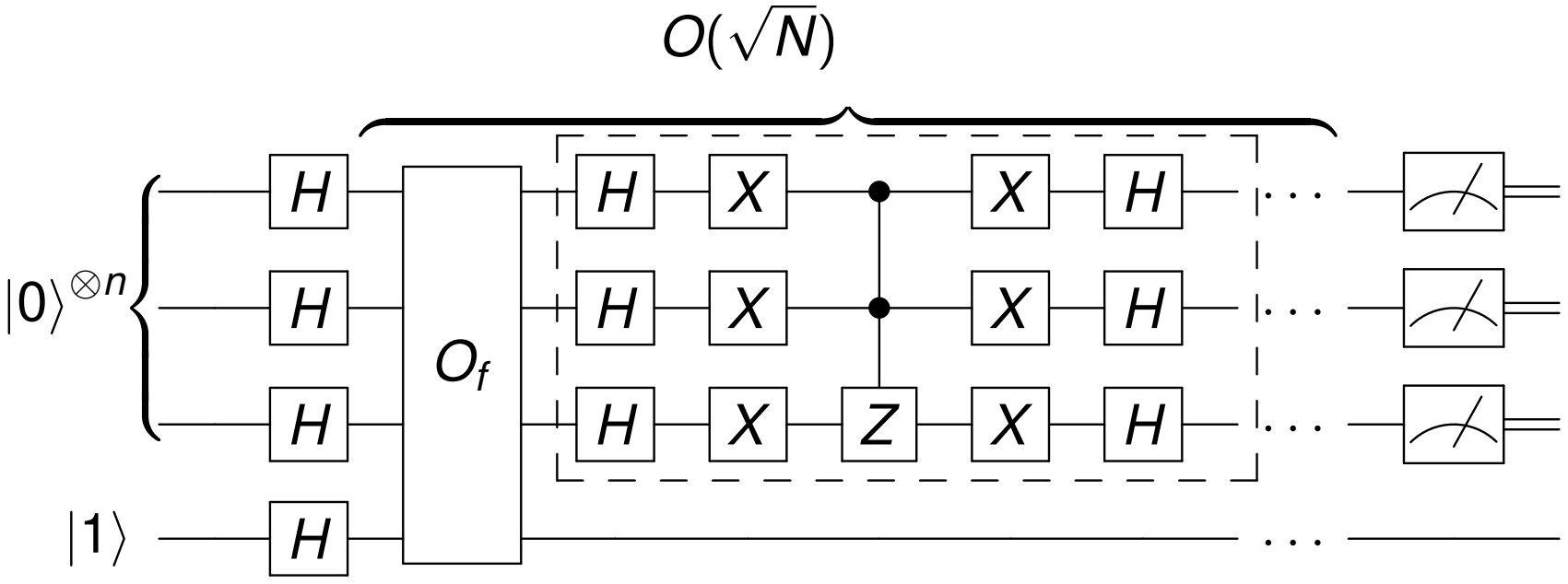
Average of all Amplitudes



Flip all Amplitudes around Avg

Grover's algorithm

- Grover's algorithm performs $O(\sqrt{N})$ iterations, each one consisting in an oracle query and a call to Grover's diffusion operator
- The oracle "marks" those states that verify the condition
- The diffusion operator "amplifies" the amplitudes of the marked states



Grover's algorithm as a rotation

- Let us denote by $|x_1\rangle$ the marked element
- Then, the initial state of the upper n qubits is

$$\sqrt{\frac{N-1}{N}}\,|x_0\rangle + \sqrt{\frac{1}{N}}\,|x_1\rangle$$

where

$$|x_0\rangle = \sum_{x\in\{0,1\}^n, x\neq x_1} \sqrt{\frac{1}{N-1}}\,|x\rangle$$

- We can choose $\theta \in (0, \frac{\pi}{2})$ such that

$$\cos\theta = \sqrt{\frac{N-1}{N}} \qquad \sin\theta = \sqrt{\frac{1}{N}}$$

Grover's algorithm as a rotation (2)

- Define D to be Grover's diffusion operator and $G = DO_f$
- It can be shown that G acts on the 2-dimensional space spawned by $|x_0\rangle$ and $|x_1\rangle$ as a rotation of angle 2θ
- That is

$$G|x_0\rangle = \cos 2\theta\,|x_0\rangle + \sin 2\theta\,|x_1\rangle$$

$$G|x_1\rangle = -\sin 2\theta\,|x_0\rangle + \cos 2\theta\,|x_1\rangle$$

$$|x_0\rangle = \sum_{x\in\{0,1\}^n, x\neq x_1} \sqrt{\frac{1}{N-1}}\,|x\rangle$$

- Since the initial state is $\cos\theta\,|x_0\rangle + \sin\theta\,|x_1\rangle$, after m iterations we will have

$$\cos(2m+1)\theta\,|x_0\rangle + \sin(2m+1)\theta\,|x_1\rangle$$

Grover's algorithm as a rotation (3)

- In order to obtain $|x_1\rangle$ with high probability when we measure we need

$$(2m+1)\theta \approx \frac{\pi}{2}$$

and this gives

$$m \approx \frac{\pi}{4\theta} - \frac{1}{2}$$

- Since

$$\sin\theta = \sqrt{\frac{1}{N}}$$

we will have

$$\theta \approx \sqrt{\frac{1}{N}}$$

and then we can choose

$$m = \left\lfloor \frac{\pi}{4}\sqrt{N} \right\rfloor$$

The case with multiple marked elements

- If the number of marked elements is $k > 1$, a similar argument can be made by defining

$$|x_0\rangle = \sum_{f(x)=0} \sqrt{\frac{1}{N-k}}\,|x\rangle$$

$$|x_1\rangle = \sum_{f(x)=1} \sqrt{\frac{1}{k}}\,|x\rangle$$

- In this case

$$\sin\theta = \sqrt{\frac{k}{N}}$$

and if $k \ll N$ we can choose

$$m = \left\lfloor \frac{\pi}{4}\sqrt{\frac{N}{k}} \right\rfloor$$

The case with unknown number of marked elements

- If we do not know how many elements are marked, we can still user Grover's algorithm
- We can use Grover's circuit combined with the Quantum Fourier Transform to estimate k
- Or we can choose m at random. For instance:
  - Uniformly from the set $\{0, \ldots, \left\lceil \sqrt{N} + 1 \right\rceil\}$
  - With an incremental scheme, starting with an upper bound for m of $b = 1$ and increasing it exponentially up to $\sqrt{N}$
- In all the cases, it can be shown that a marked element will be found with high probability with $O(\sqrt{N})$ queries to the oracle

Some comments on Grover's algorithm

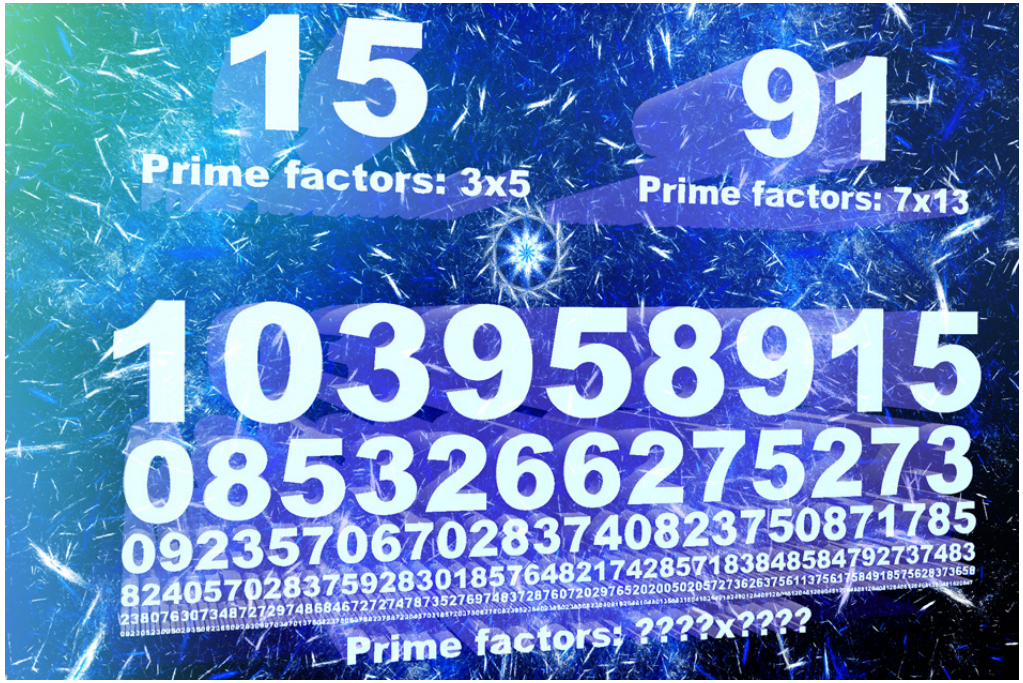
- When we measure, we will obtain x such that $f(x) = 1$ with probability depending on:
  - The number m of iterations
  - The fraction of values x that satisfy the condition
- If we perform too many iterations, we can overshoot and not find a marked element
- On the other hand, if $k = \frac{N}{4}$ then one iteration will find a marked element with certainty
- Grover's algorithm can be used to find minima of functions (Dürr-Hoyer's algorithm)
- It can be shown that no other quantum algorithm can obtain more than a quadratic speed-up over over classical algorithms in the same setting
- A generalization of Grover's algorithm called Amplitude Amplification can be used with states prepared by an arbitrary unitary A

Part XIV

Shor's algorithm: breaking the Internet

Shor's algorithm and factoring

- Shor's algorithm is, probably, the most famous quantum algorithm
- It finds a factor of a n-bit integer in time $O(n^2(\log n)(\log\log n))$
- The best classical algorithm that we know of for the same task needs time $O(e^{cn^{\frac{1}{3}}(\log n)^{\frac{2}{3}}})$
- Dramatic consequences for current cryptography (RSA)



Steps of Shor's algorithm

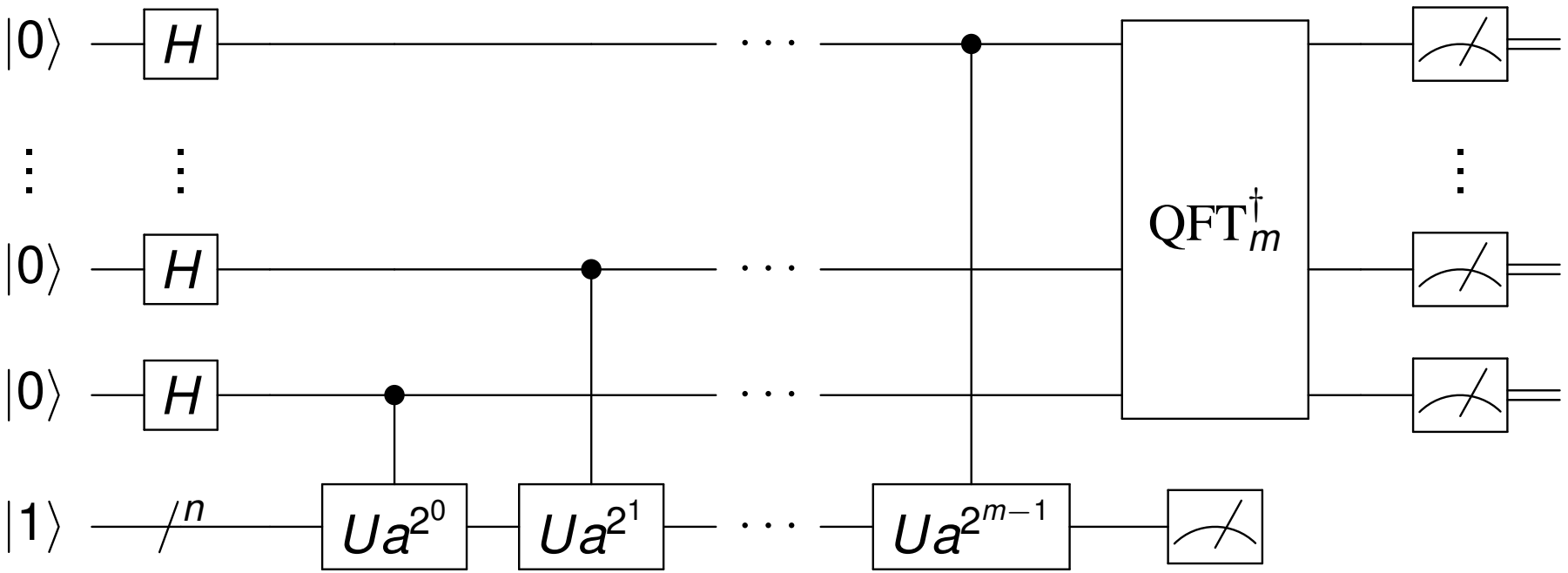
1. Given N, check that N is not a prime or power of a prime. If it is, stop.
2. Choose $1 < a < N$ at random
3. If $b = gcd(a, N) > 1$, output b and stop
4. Find the order of a mod N, that is, $r > 0$ such that $a^r \equiv 1 \mod N$
5. If r is odd, go to 2
6. Compute
$$x = a^{\frac{r}{2}} + 1 \mod N$$
$$y = a^{\frac{r}{2}} - 1 \mod N$$
7. If $x = 0$, go to 2. If $y = 0$, take $r = \frac{r}{2}$ and go to 5.
8. Compute $p = gcd(x, N)$ and $q = gcd(y, N)$. At least one of them will be a non-trivial factor of N

Correctness of Shor's algorithm

- We know that
$$a^r \equiv 1 \mod N$$
- Thus
$$x \cdot y \equiv (a^{\frac{r}{2}} + 1)(a^{\frac{r}{2}} - 1) \equiv (a^r - 1) \equiv 0 \mod N$$
- This means that $x \cdot y$ is a multiple of N
- Since neither x nor y are multiples of N, either p or q divides N
- It can be proved that step 8 will be reached with high probability

Implementation of Shor's algorithm

- Every step but number 4 are carried out on a classical computer (efficient algorithms exist)
- For step 4, there exists a quantum circuit with a number of gates that is polynomial on n (the number of bits of N)



Preparing a periodic sequence

- The first part of the circuit computes

$$\frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x\rangle \, |a^x \mod N\rangle$$

- When we measure the bottom qubits, we obtain

$$\frac{1}{\sqrt{|C|}} \sum_{x \in C} |x\rangle \, |c\rangle$$

where c is some value in $\{0, \ldots, N-1\}$ and $C = \{x : a^x \mod N = c\}$.

Preparing a periodic sequence (2)

- For example, if $a = 2$, $N = 5$, $m = 4$, we would have

  $$\frac{1}{4}\left(|0\rangle|1\rangle + |1\rangle|2\rangle + |2\rangle|4\rangle + |3\rangle|3\rangle + |4\rangle|1\rangle + \ldots + |15\rangle|3\rangle\right)$$

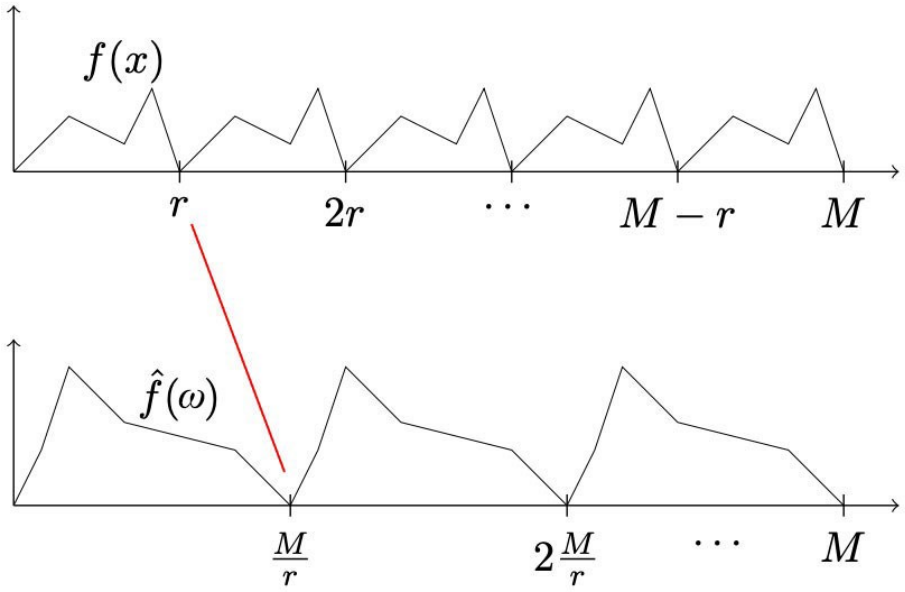
  and when we measure we could obtain, for instance

  $$\frac{1}{2}\left(|1\rangle|2\rangle + |5\rangle|2\rangle + |9\rangle|2\rangle + |13\rangle|2\rangle\right)$$

- Notice that the values of the first register are exactly 4 units apart and that $2^4 = 1 \mod 5$.
- In general, we will obtain values that are r units apart, where $a^r = 1 \mod N$.

Measuring the period

- To retrieve the period r we use the (inverse) of the Quantum Fourier Transform (QFT)
- Two properties of the QFT are central here:
  - Shift-invariance (up to an unobservable phase)
  - QFT transforms sequences with period r into sequences with period $\frac{M}{r}$ (where $M = 2^m$)
- After the use of the inverse QFT, we can measure a value of the form $\frac{Mc}{r}$ with high probability and, from it, obtain r

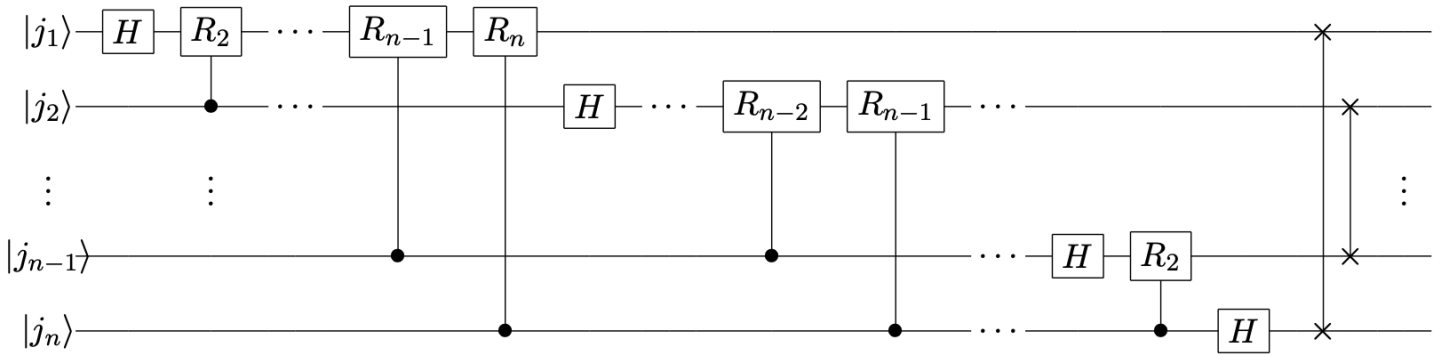


Quantum Fourier Transform: definition and circuit

- The QFT of order m is the unitary transformation defined by

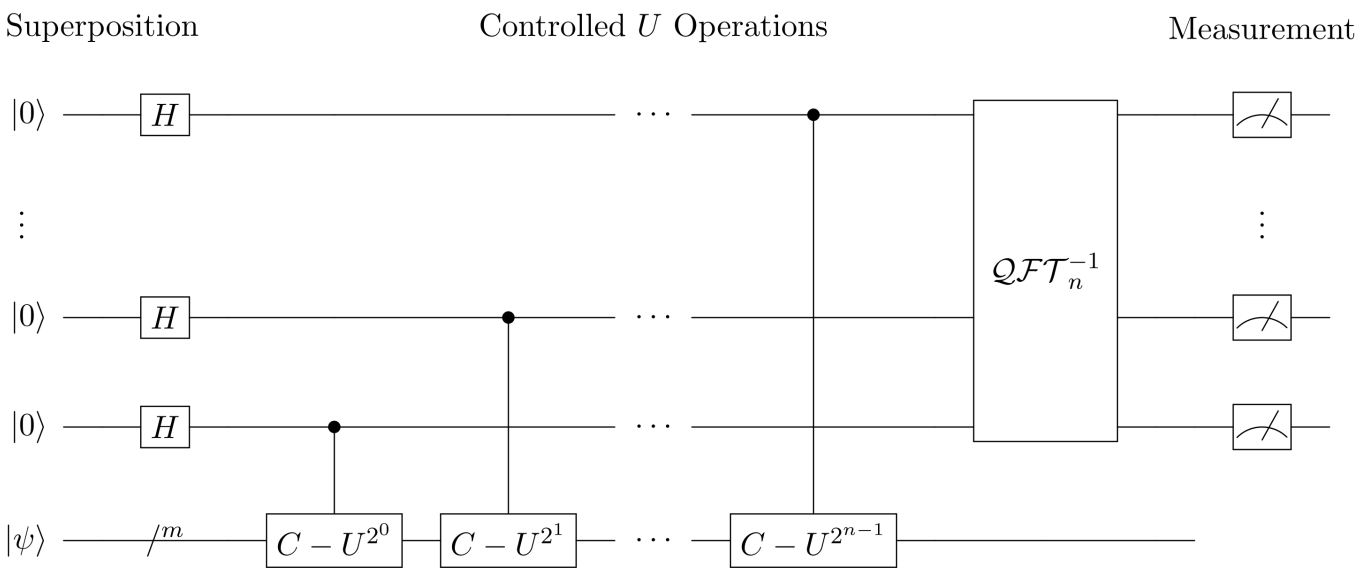
$$QFT\,|j\rangle = \frac{1}{\sqrt{2^m}} \sum_{k=0}^{2^m-1} e^{\frac{2\pi ijk}{2^m}} |k\rangle$$

- The circuit in the figure implements the QFT
- The R_k gates in the circuit are what we call $R_Z(\frac{2\pi}{2^k})$
- The number of gates is quadratic in m, an exponential speed-up over the classical case (FFT)
- For Shor, m can be chosen to be about $2n$



Using the QFT for phase estimation

- Suppose we are given a unitary operation U and one of its eigenvectors $|\psi\rangle$
- We know that there exists $\theta \in [0, 1)$ such that $U|\psi\rangle = e^{2\pi i\theta}$
- We can estimate θ with the circuit shown below
- With the first part, we will obtain $\frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i\theta k} |k\rangle$
- By using the inverse QFT we can measure $j \approx 2^n\theta$



Shor's algorithm as a particular case of quantum phase estimation

- Clearly, the circuit used in Shor's algorithm is a case of quantum phase estimation
- It can be shown that the (unitary) operation of modular mutiplication by a has eigenvalues

$$e^{2\pi i \frac{k}{r}} \qquad k = 0, \ldots, r-1$$

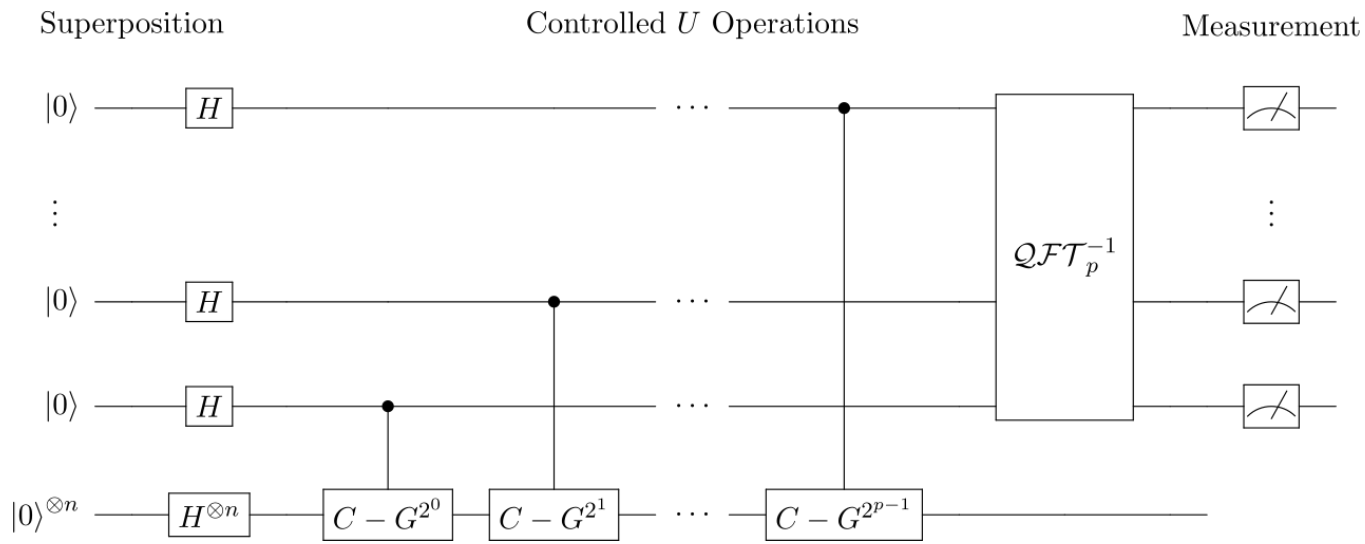
  where r is the period of a
- It is not easy to prepare one of the eigenvectors $|\psi_k\rangle$ of the unitary operation
- But we use the fact that

$$|1\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |\psi_k\rangle$$

- We will then measure a value close to $\frac{2^m k}{r}$ for some k

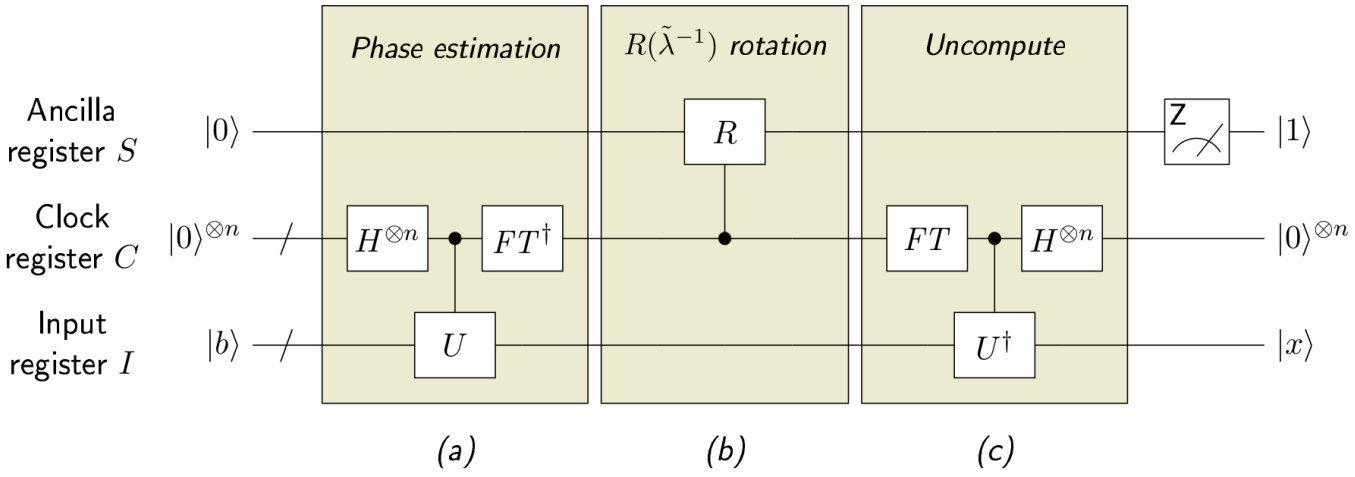
Using quantum phase estimation to count the number of marked elements

- We can use Grover's algorithm together with the QFT to count the number of elements marked by a boolean function
- The eigenvalues of Grover's operator are $e^{\pm 2i\theta}$ where $\sin\theta = \sqrt{\frac{k}{N}}$
- Then, with quantum phase estimation we can recover k, the number of marked elements



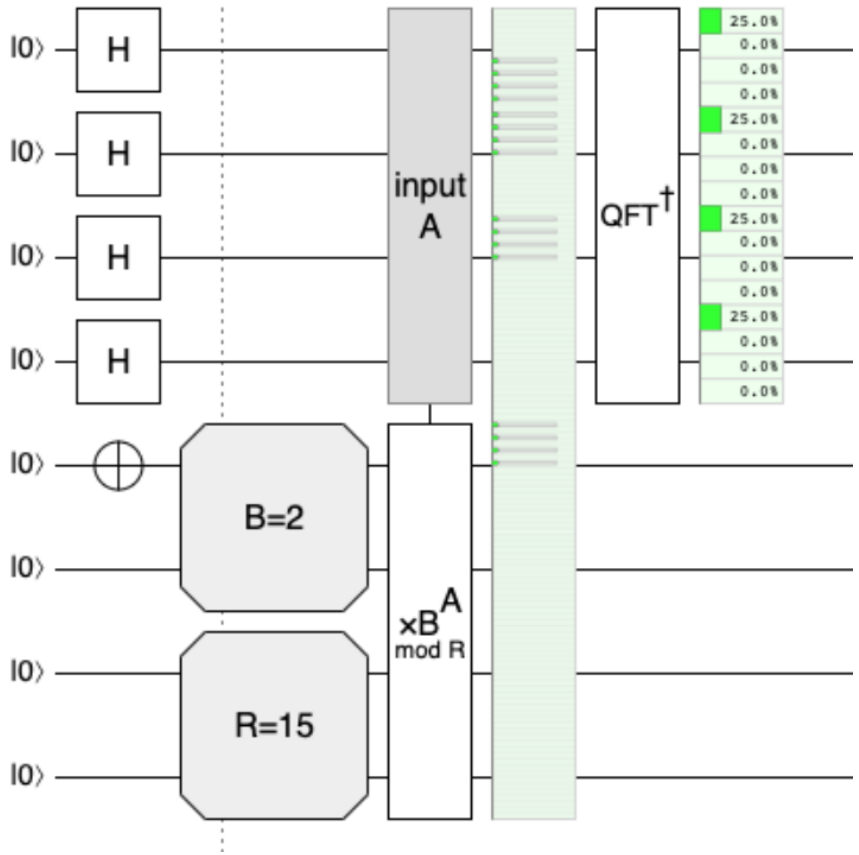
HHL: Applying quantum phase estimations to solve linear systems of equations

- A quantum algorithm proposed in 2009 by Harrow, Hassidim and Lloyd can be used to solve linear systems of equations
- The main steps of the algorithm are
  - Computation of the eigenvalues (quantum phase estimation)
  - Inversion of the eigenvalues
  - Uncomputation of the eigenvalues (inverse of quantum phase estimation)



Visualizing Shor's algorithm with Qirk

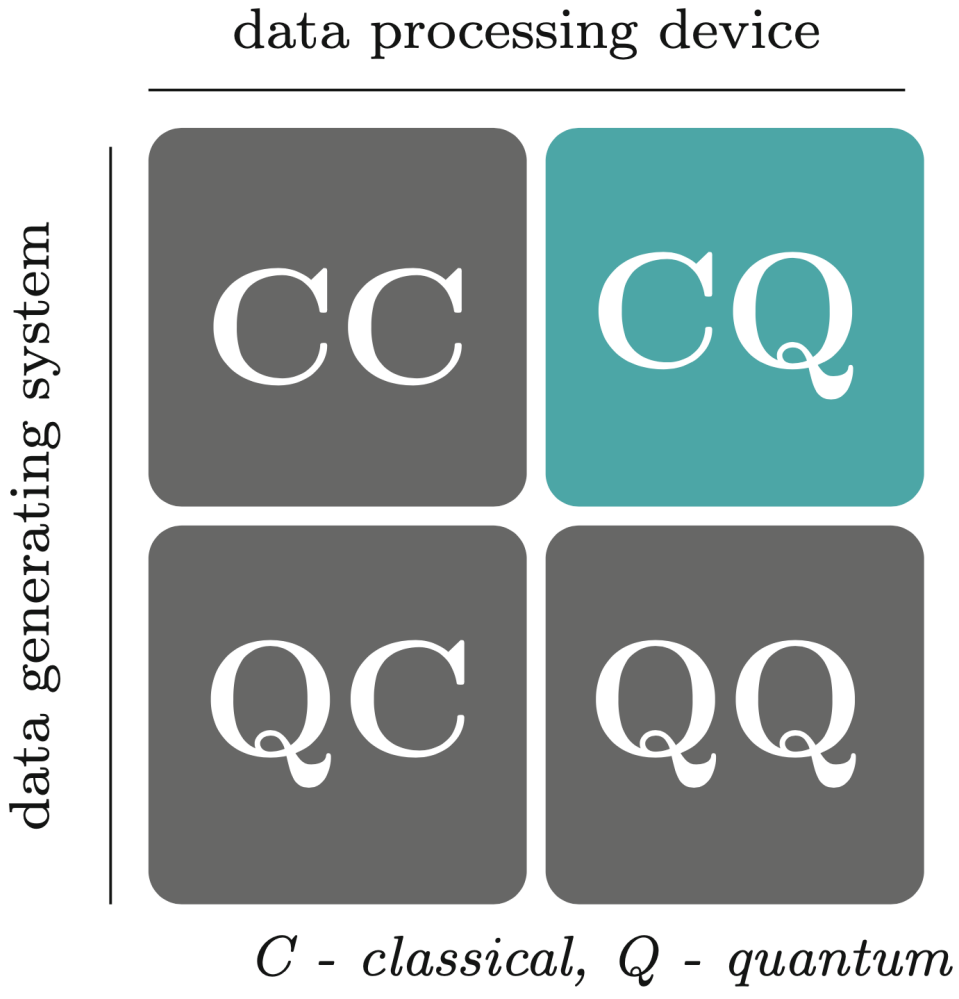
- Case $a = 2$ and $N = 15$
- Case $a = 4$ and $N = 15$
- Case $a = 14$ and $N = 15$
- Case $a = 26$ and $N = 55$



Part XV

Quantum Machine Learning: a marriage made in heaven

What I talk about when I talk about Quantum Machine Learning

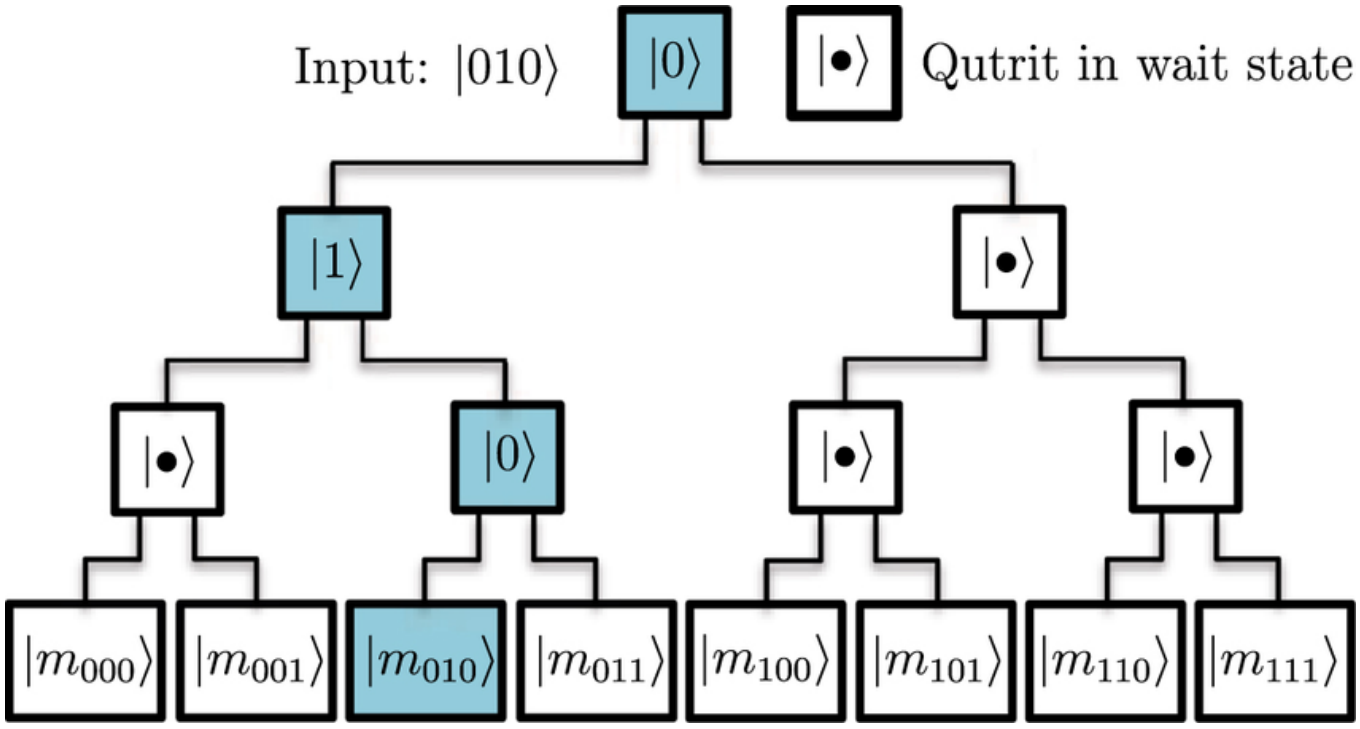


QBLAS: The Quantum Basic Linear Algebra Subroutines

- A number of algorithms in Quantum Machine Learning (QML) rely on the exponential speedup of methods such as
  - Quantum Fourier Transform
  - Quantum Phase Estimation
  - HHL
- We refer to these methods as Quantum Basic Linear Algebra Subroutines (QBLAS)
- Other quantum subroutines used in QML include amplitude amplification and quantum annealing
- Some common problems are how to load the **input**, how to read the **output** and the **size** of the circuits

QRAM: The elephant in the room

- A Quantum Random Access Memory should allow queries in superposition
- Several architectures have been proposed (for instance, the “bucket brigade”) but further investigation is needed
- Loading data can become a bottleneck for many QML algorithms



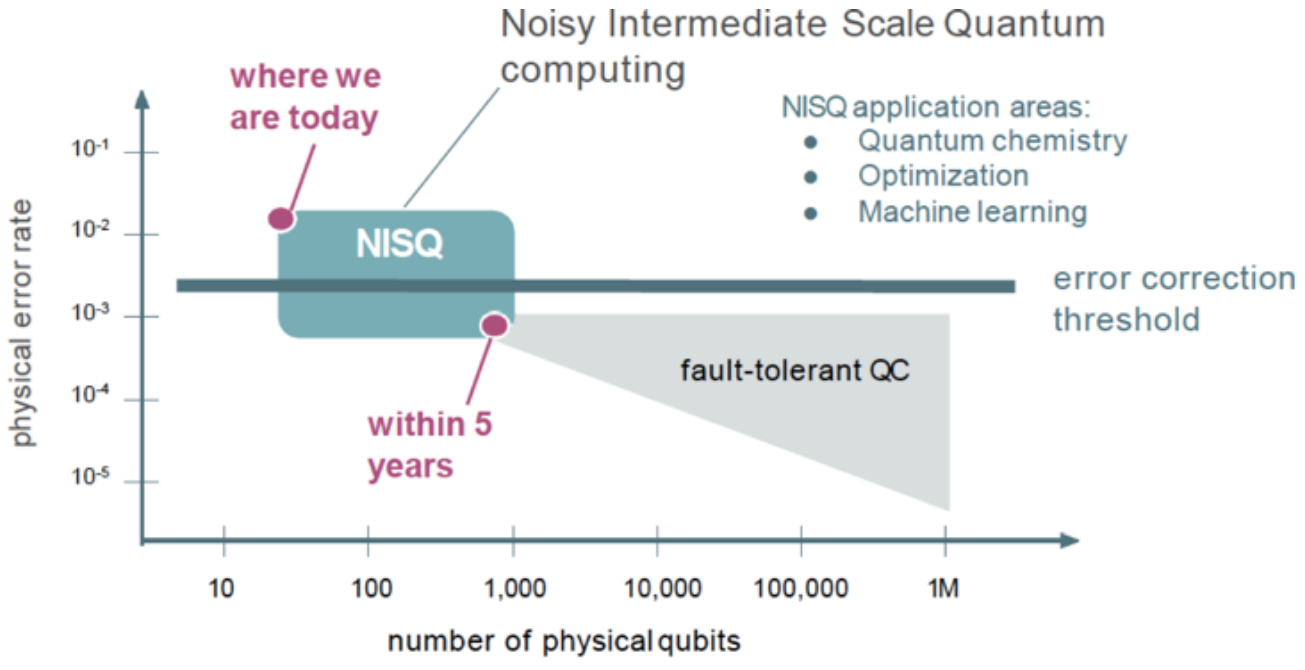
Translational QML and speedups

Method	Speedup	Amplitude amplification	HHL	Adiabatic	qRAM
Bayesian inference[106,107]	$O(\sqrt{N})$	Yes	Yes	No	No
Online perceptron[108]	$O(\sqrt{N})$	Yes	No	No	Optional
Least-squares fitting[9]	$O(\log N)$*	Yes	Yes	No	Yes
Classical Boltzmann machine[20]	$O(\sqrt{N})$	Yes/No	Optional/ No	No/Yes	Optional
Quantum Boltzmann machine[22,61]	$O(\log N)$*	Optional/No	No	No/Yes	No
Quantum PCA[11]	$O(\log N)$*	No	Yes	No	Optional
Quantum support vector machine[13]	$O(\log N)$*	No	Yes	No	Yes
Quantum reinforcement learning[30]	$O(\sqrt{N})$	Yes	No	No	No

*There exist important caveats that can limit the applicability of the method[51].

QML in the times on NISQ

- Noisy Intermediate-Scale Quantum computers are
  - Subjet to noise (not fault-tolerant)
  - Limited in the number of qubits (50-100)
  - Not fully-connected
- Despite these drawbacks, they may be useful for QML



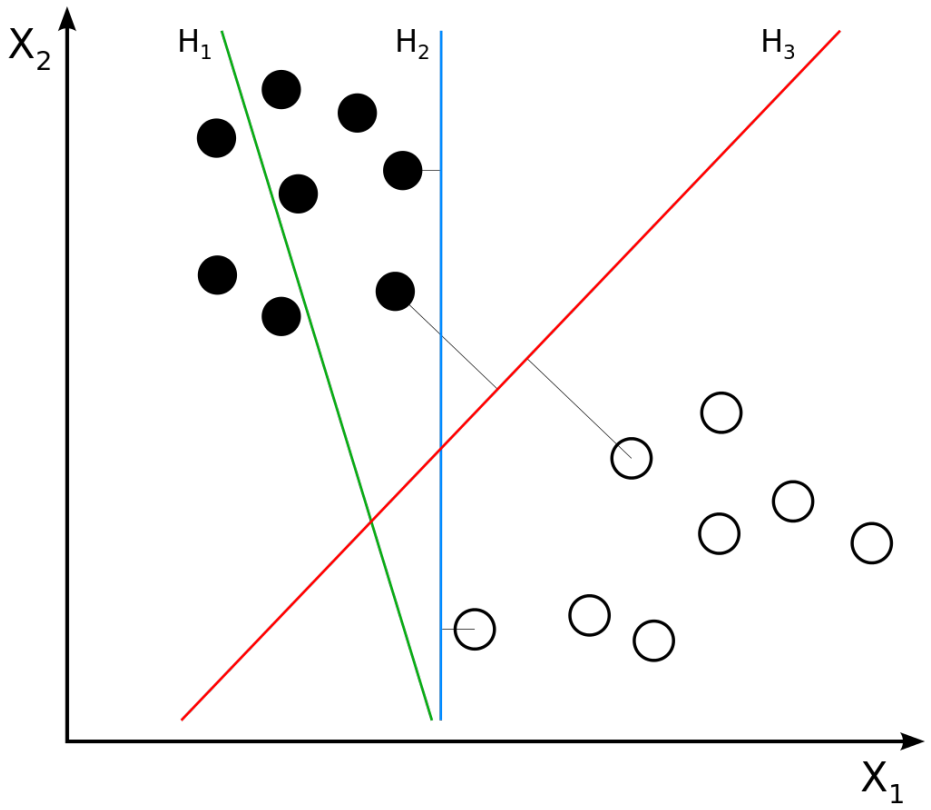
"Quantum computing in the NISQ era and beyond" Preskill, 2018 https://arxiv.org/abs/1801.00862

Part XVI

Quantum Support Vector Machines: exploiting the kernel trick

Support Vector Machines

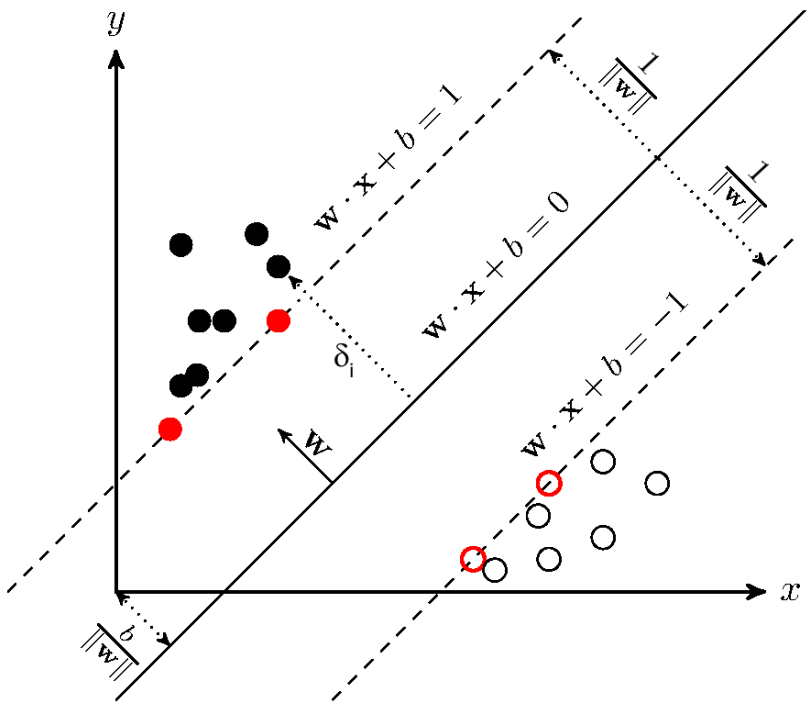
- Support Vector Machines (SVM) are a very popular machine learning algorithm used for data classification
- The main idea is to find a hyperplane that separates data from two different classes with the maximum possible margin



Finding the hyperplane

- We are given training data points (x_i, y_i) where the x_i are vectors of real numbers and $y_i \in \{1, -1\}$
- The problem of finding the separating hyperplane with the biggest margin can be formulated as

$$\text{Minimize } \frac{1}{2}||w||^2 \text{ subject to } y_i(w \cdot x_i + b) \geq 1$$



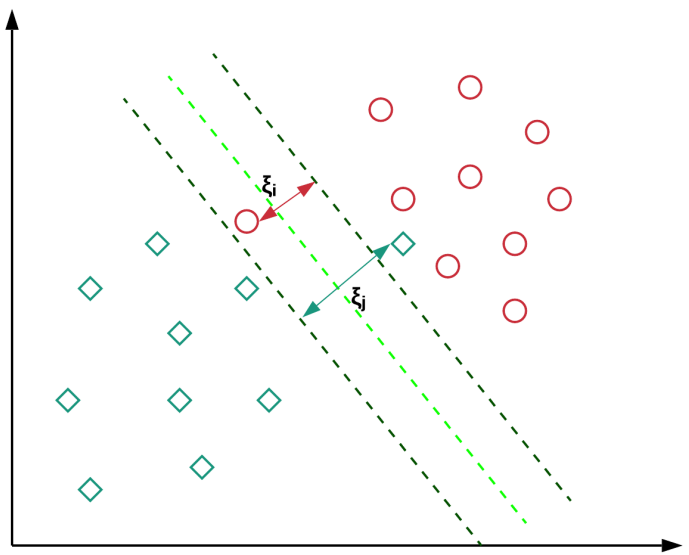
The soft-margin case

- In the "soft-margin" case we introduce a hyperparameter $C \geq 0$ and reformulate the problem as

$$\text{Minimize } \frac{1}{2}||w||^2 + C\sum_i \xi_i$$

subject to

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0$$



Dual formulation of SVM

- An equivalent formulation of the SVM optimization problem is this dual formulation

$$\text{Maximize } \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} y_i y_j \alpha_i \alpha_j \left(x_i \cdot x_j \right)$$

subject to

$$0 \leq \alpha_i \leq C \quad \sum_i \alpha_i y_i = 0$$

- From the values α_i we can recover b and w. In fact

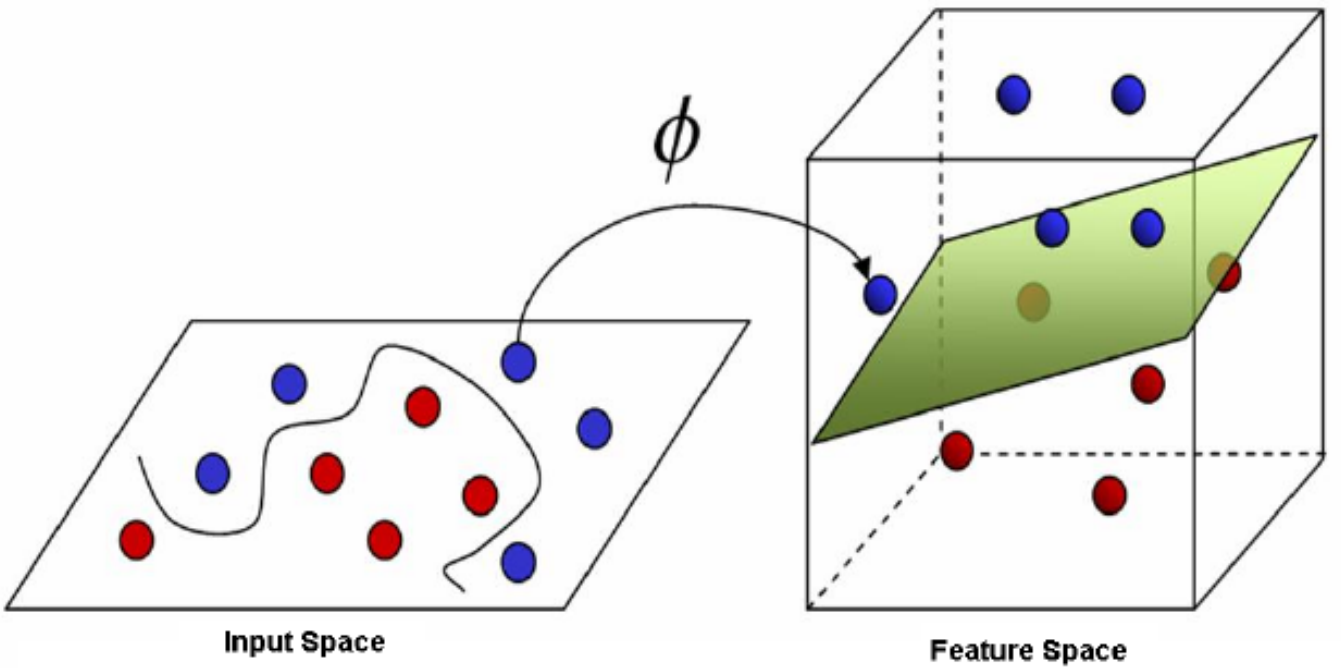
$$w = \sum_i \alpha_i y_i x_i$$

and to classify a point x we compute

$$w \cdot x + b = \sum_i \alpha_i y_i \left(x_i \cdot x \right) + b$$

Non-linear separation

- A common technique to improve classification with Support Vector Machines is to embed the data points x_i into a higher-dimensional space using a feature map $\phi(x_i)$



The Kernel Trick

- We can easily incorporate the feature map in our formulation of the dual problem for the SVM

$$\text{Maximize } \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} y_i y_j \alpha_i \alpha_j \left(\phi(x_i) \cdot \phi(x_j)\right)$$

subject to

$$0 \leq \alpha_i \leq C \quad \sum_i \alpha_i y_i = 0$$

- Again, we can obtain w as

$$w = \sum_i \alpha_i y_i \phi(x_i)$$

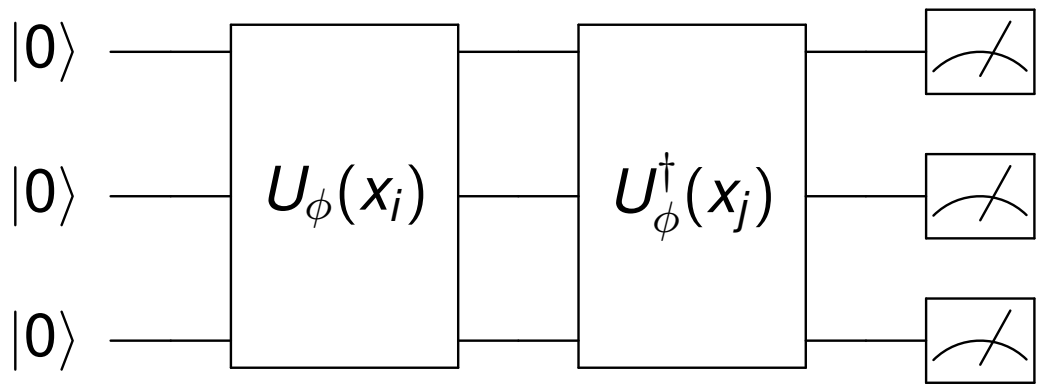
and to classify a point x we only need to compute

$$w \cdot x + b = \sum_i \alpha_i y_i \left(\phi(x_i) \cdot \phi(x)\right) + b$$

- The function $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ is called "kernel"

Computing kernel functions with quantum computers

- In 2019, Havlíček, Córcoles, Temme et al. proposed using quantum computers as kernel estimators
- Each data point x_i is embedded in a Hilbert space by means of a variational circuit $U_\phi(x_i)$ such that $U_\phi(x_i)\,|0\rangle = |\phi(x_i)\rangle$
- We know we can $|\langle\phi(x_j)|\phi(x_i)\rangle|^2$ by running the circuit of the figure and computing the relative frequency of $|0\rangle$

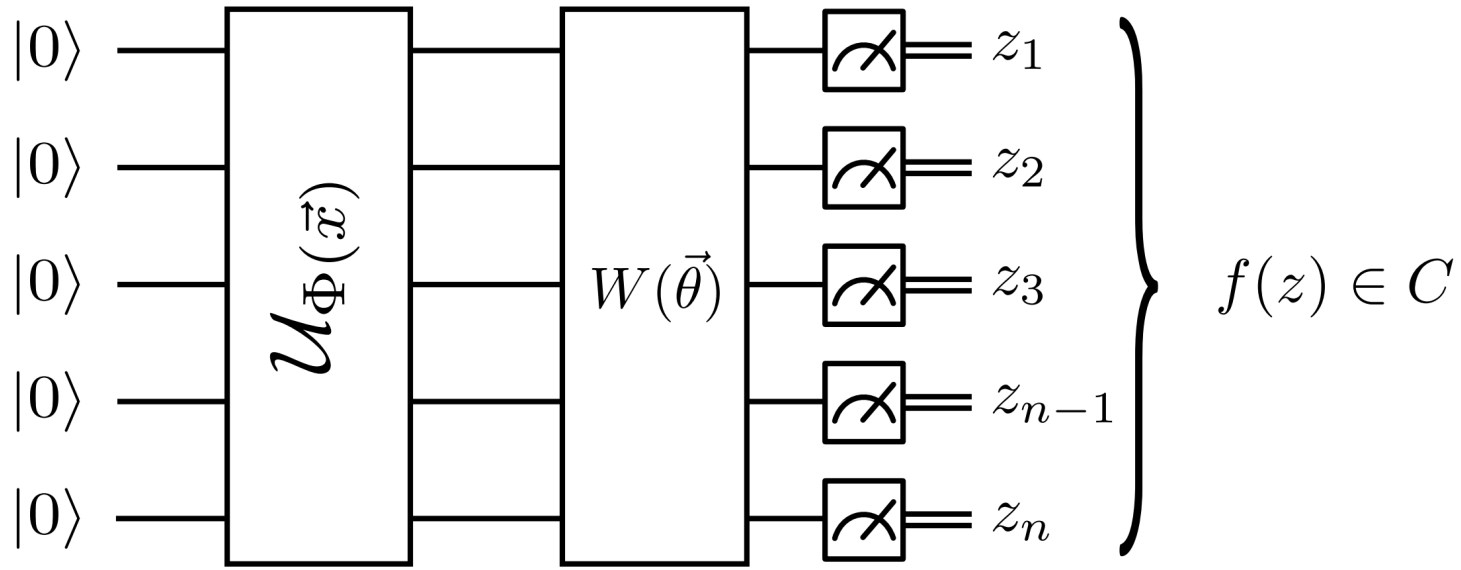


Part XVII

Quantum Neural Networks: Deep Learning meets Quantum Computing

What is a Quantum Neural Network

- Quantum Neural Networks or Variational Quantum Classifiers are parametrized quantum circuits that can be "trained" on data and used for classification tasks
- The most common architecture is shown in the figure below: a feature map that embeds the data point into the Hilbert space and a variational form that performs the classification



Training and classifying with a Quantum Neural Network

- A QNN prepares a state $|\psi(x,\theta)\rangle$ that depends on the input data x and the parameters θ
- We measure the state and compute an average value, for instance

$$f(x,\theta) = \langle\psi(x,\theta)| Z_1 \cdots Z_n |\psi(x,\theta)\rangle$$

- For each training example x_i we have a class y_i
- We choose a loss function L and we want to find θ minimizing

$$\sum_i L(y_i, f(x,\theta))$$

- Once we obtain the optimal value θ_{min} we can predict a class for x using $f(x,\theta_{min})$

Gradients and the parameter shift rule

- To obtain θ_{min}, we can use a classical minimizer
- If we need to compute gradients of f the parameter-shift rule is useful
- Suppose

$$U(\theta) = e^{-i\theta H}$$

with H a Hermitian matrix with eigenvalues $\pm r$ (r real)

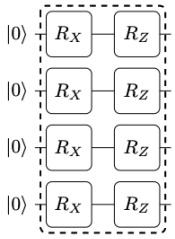
- This is the case, for instance, if U is a one-qubit rotation
- Then, we have

$$\frac{\partial f(x,\theta)}{\partial \theta} = r \cdot [f(x,\theta+s) - f(x,\theta-s)]$$

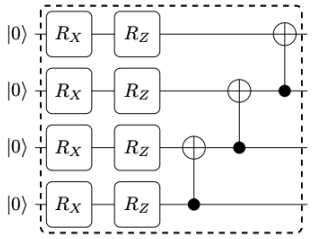
where $s = \frac{\pi}{4r}$

- This requires just two extra evaluations of the same circuit with shifted parameters

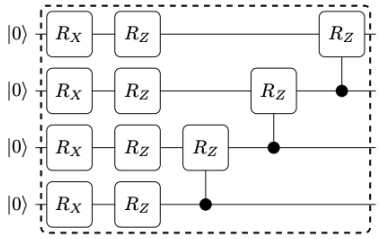
Choosing feature maps and variational forms



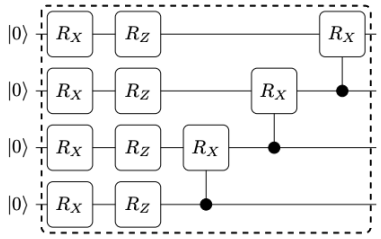
Circuit 1



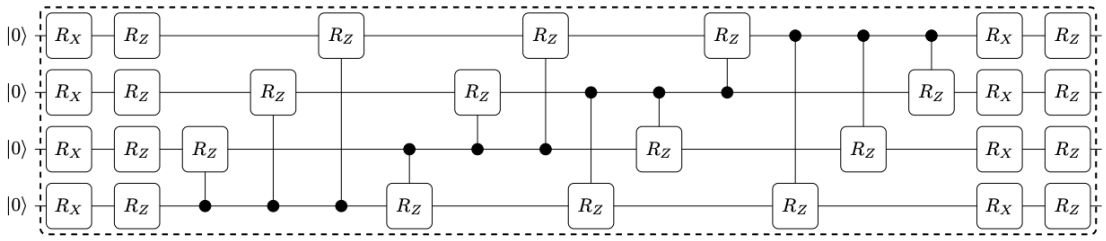
Circuit 2



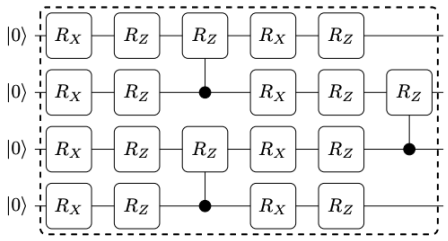
Circuit 3



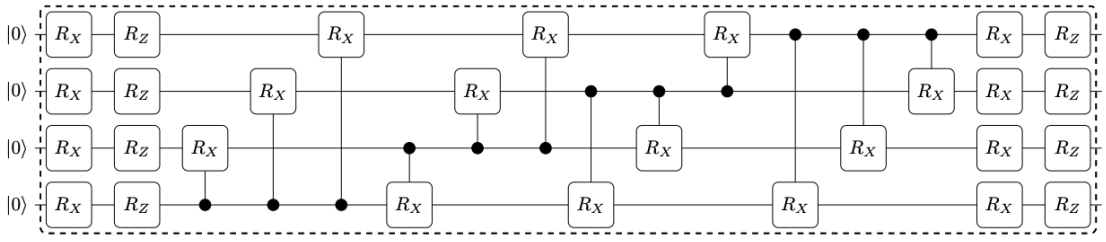
Circuit 4



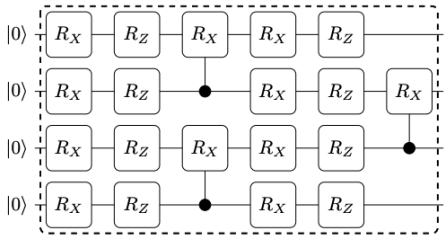
Circuit 5



Circuit 7

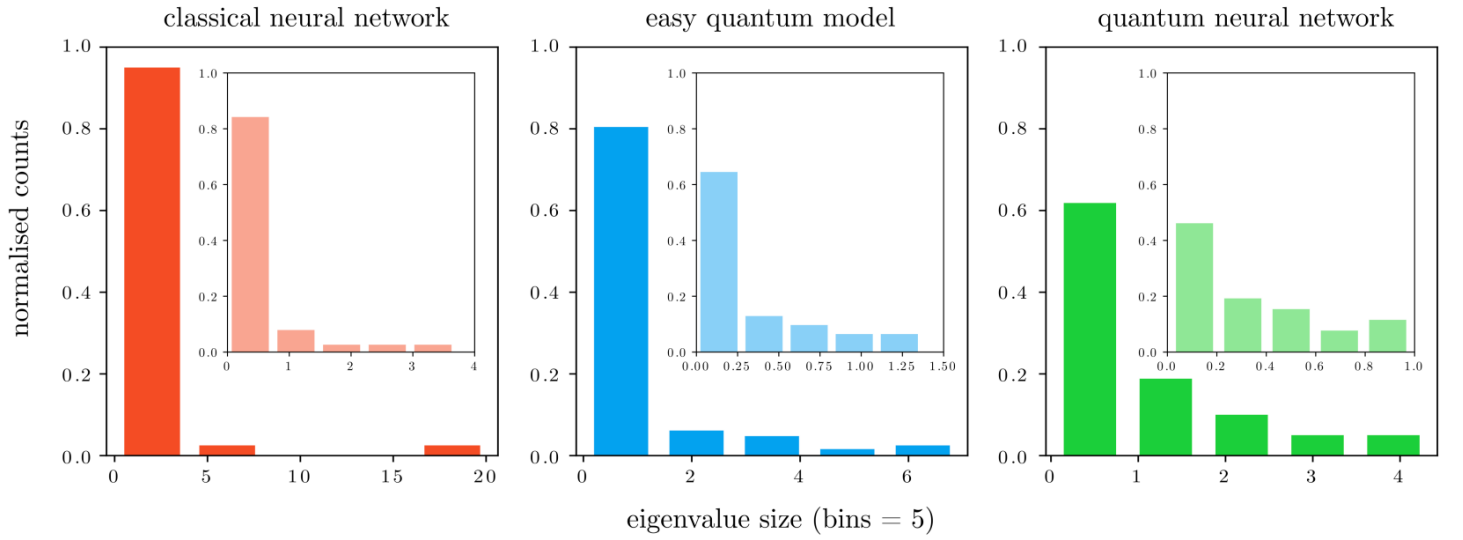


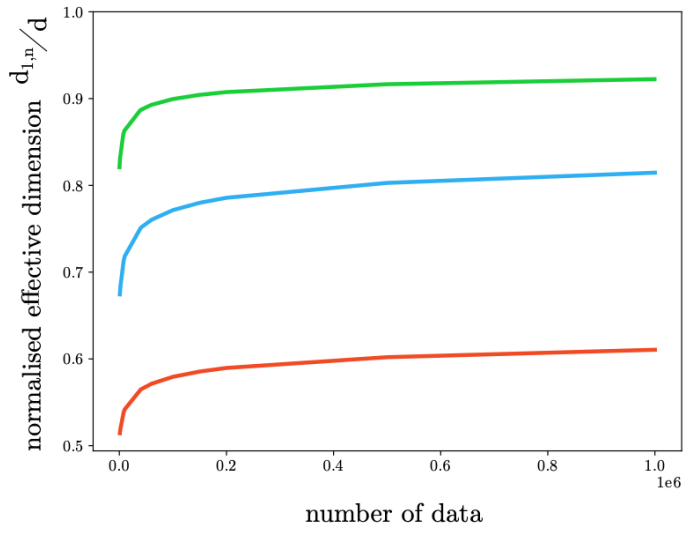
Circuit 6

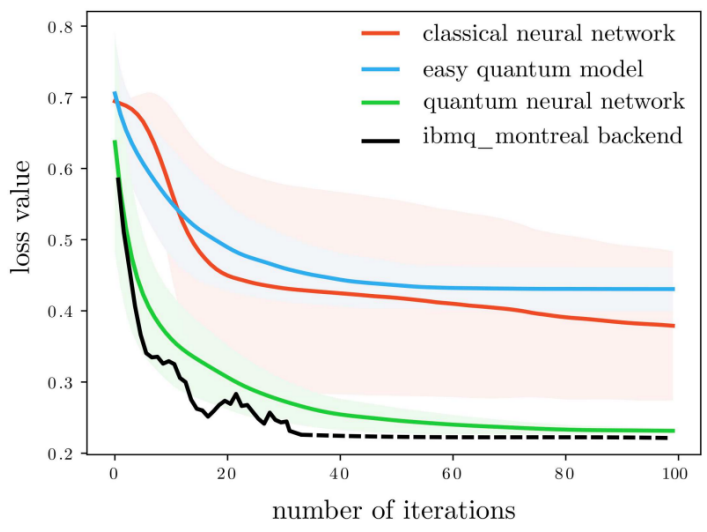


Circuit 8

The power of quantum neural networks







Abbas, A., Sutter, D., Zoufal, C. *et al.* The power of quantum neural networks. *Nat Comput Sci* **1**, 403–409 (2021). https://doi.org/10.1038/s43588-021-00084-1

Advantage in practice?

Better than classical? The subtle art of benchmarking quantum machine learning models

Joseph Bowles,[1,*] Shahnawaz Ahmed,[1,2,†] and Maria Schuld[1,‡]

[1] *Xanadu, Toronto, ON, M5G 2C8, Canada*
[2] *Chalmers University of Technology*

(Dated: March 15, 2024)

Benchmarking models via classical simulations is one of the main ways to judge ideas in quantum machine learning before noise-free hardware is available. However, the huge impact of the experimental design on the results, the small scales within reach today, as well as narratives influenced by the commercialisation of quantum technologies make it difficult to gain robust insights. To facilitate better decision-making we develop an open-source package based on the PennyLane software framework and use it to conduct a large-scale study that systematically tests 12 popular quantum machine learning models on 6 binary classification tasks used to create 160 individual datasets. We find that overall, out-of-the-box classical machine learning models outperform the quantum classifiers. Moreover, removing entanglement from a quantum model often results in as good or better performance, suggesting that "quantumness" may not be the crucial ingredient for the small learning tasks considered here. Our benchmarks also unlock investigations beyond simplistic leaderboard comparisons, and we identify five important questions for quantum model design that follow from our results.

Much has been written about the "potential" of quantum machine learning, a discipline that asks how quantum computers fundamentally change what we can learn from data [1, 2]. While we have no means of running quantum algorithms on noise-free hardware yet, there are only a limited number of tools available to assess this potential. Besides proving advantages for artificial problem settings on paper, certain ideas – most prominently, variational models designed for near-term quantum technologies – can be tested in classical simulations using small datasets. Such benchmarks have in fact become a standard practice in the quantum machine learning literature and are found in almost every paper.

A taste for the results derived from small-scale benchmarks can be obtained through a simple literature review exercise. Out of 55 relevant papers published on the preprint server *arXiv*[1] until December 2023 that con-

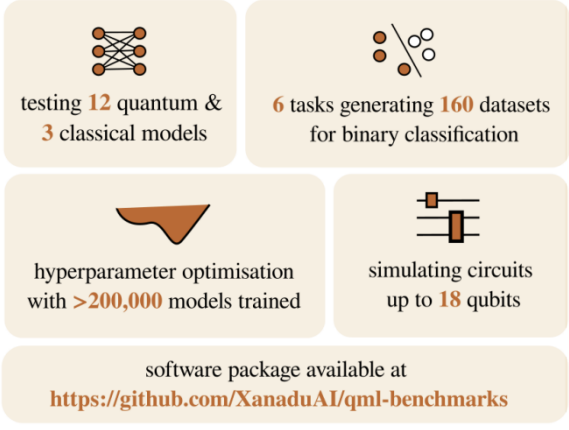


FIG. 1. The scope of the benchmark study at a glance.

Backpropagation

On quantum backpropagation, information reuse, and cheating measurement collapse

Amira Abbas[1,2,3,4], Robbie King[5], Hsin-Yuan Huang[5,6], William J. Huggins[1], Ramis Movassagh[1], Dar Gilboa[1], and Jarrod R. McClean[1*]

[1] *Google Quantum AI, Venice, California 90291, USA*
[2] *University of KwaZulu-Natal, Durban, South Africa*
[3] *Institute of Physics, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands*
[4] *QuSoft, CWI, Science Park 123, 1098 XG Amsterdam, The Netherlands*
[5] *Department of Computing and Mathematical Sciences, Caltech, Pasadena, CA 91125, USA*
[6] *Institute for Quantum Information and Matter, Caltech, Pasadena, CA 91125, USA*

Abstract

The success of modern deep learning hinges on the ability to train neural networks at scale. Through clever reuse of intermediate information, backpropagation facilitates training through gradient computation at a total cost roughly proportional to running the function, rather than incurring an additional factor proportional to the number of parameters – which can now be in the trillions. Naively, one expects that quantum measurement collapse entirely rules out the reuse of quantum information as in backpropagation. But recent developments in shadow tomography, which assumes access to multiple copies of a quantum state, have challenged that notion. Here, we investigate whether parameterized quantum models can train as efficiently as classical neural networks. We show that achieving backpropagation scaling is impossible without access to multiple copies of a state. With this added ability, we introduce an algorithm with foundations in shadow tomography that matches backpropagation scaling in quantum resources while reducing classical auxiliary computational costs to open problems in shadow tomography. These results highlight the nuance of reusing quantum information for practical purposes and clarify the unique difficulties in training large quantum models, which could alter the course of quantum machine learning.

Barren plateaus

ARTICLE

DOI: 10.1038/s41467-018-07090-4 OPEN

Barren plateaus in quantum neural network training landscapes

Jarrod R. McClean[1], Sergio Boixo[1], Vadim N. Smelyanskiy[1], Ryan Babbush[1] & Hartmut Neven[1]

Many experimental proposals for noisy intermediate scale quantum devices involve training a parameterized quantum circuit with a classical optimization loop. Such hybrid quantum-classical algorithms are popular for applications in quantum simulation, optimization, and machine learning. Due to its simplicity and hardware efficiency, random circuits are often proposed as initial guesses for exploring the space of quantum states. We show that the exponential dimension of Hilbert space and the gradient estimation complexity make this choice unsuitable for hybrid quantum-classical algorithms run on more than a few qubits. Specifically, we show that for a wide class of reasonable parameterized quantum circuits, the probability that the gradient along any reasonable direction is non-zero to some fixed precision is exponentially small as a function of the number of qubits. We argue that this is related to the 2-design characteristic of random circuits, and that solutions to this problem must be studied.

ARTICLE

Check for updates

https://doi.org/10.1038/s41467-021-21728-w OPEN

Cost function dependent barren plateaus in shallow parametrized quantum circuits

M. Cerezo[1,2✉], Akira Sone[1,2], Tyler Volkoff[1], Lukasz Cincio[1] & Patrick J. Coles[1✉]

Variational quantum algorithms (VQAs) optimize the parameters $\boldsymbol{\theta}$ of a parametrized quantum circuit $V(\boldsymbol{\theta})$ to minimize a cost function C. While VQAs may enable practical applications of noisy quantum computers, they are nevertheless heuristic methods with unproven scaling. Here, we rigorously prove two results, assuming $V(\boldsymbol{\theta})$ is an alternating layered ansatz composed of blocks forming local 2-designs. Our first result states that defining C in terms of global observables leads to exponentially vanishing gradients (i.e., barren plateaus) even when $V(\boldsymbol{\theta})$ is shallow. Hence, several VQAs in the literature must revise their proposed costs. On the other hand, our second result states that defining C with local observables leads to at worst a polynomially vanishing gradient, so long as the depth of $V(\boldsymbol{\theta})$ is $\mathcal{O}(\log n)$. Our results establish a connection between locality and trainability. We illustrate these ideas with large-scale simulations, up to 100 qubits, of a quantum autoencoder implementation.

Defining new architectures

PHYSICAL REVIEW X **11,** 041011 (2021)

Absence of Barren Plateaus in Quantum Convolutional Neural Networks

Arthur Pesah,[1,2] M. Cerezo,[1,3] Samson Wang,[1,4] Tyler Volkoff,[1] Andrew T. Sornborger,[5] and Patrick J. Coles[1]

[1]*Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*
[2]*Department of Physics and Astronomy, University College London, London WC1E 6BT, United Kingdom*
[3]*Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, New Mexico 87544*
[4]*Imperial College London, London, United Kingdom*
[5]*Information Sciences, Los Alamos National Laboratory, Los Alamos, New Mexico 87544, USA*

(Received 12 March 2021; revised 13 July 2021; accepted 2 August 2021; published 15 October 2021)

Quantum neural networks (QNNs) have generated excitement around the possibility of efficiently analyzing quantum data. But this excitement has been tempered by the existence of exponentially vanishing gradients, known as barren plateau landscapes, for many QNN architectures. Recently, quantum convolutional neural networks (QCNNs) have been proposed, involving a sequence of convolutional and pooling layers that reduce the number of qubits while preserving information about relevant data features. In this work, we rigorously analyze the gradient scaling for the parameters in the QCNN architecture. We find that the variance of the gradient vanishes no faster than polynomially, implying that QCNNs do not exhibit barren plateaus. This result provides an analytical guarantee for the trainability of randomly initialized QCNNs, which highlights QCNNs as being trainable under random initialization unlike many other QNN architectures. To derive our results, we introduce a novel graph-based method to analyze expectation values over Haar-distributed unitaries, which will likely be useful in other contexts. Finally, we perform numerical simulations to verify our analytical results.

DOI: 10.1103/PhysRevX.11.041011 Subject Areas: Quantum Information

Simulability

Does provable absence of barren plateaus imply classical simulability? Or, why we need to rethink variational quantum computing

M. Cerezo,[1,2,*] Martin Larocca,[3,4] Diego García-Martín,[1] N. L. Diaz,[1,5] Paolo Braccia,[3] Enrico Fontana,[6] Manuel S. Rudolph,[7] Pablo Bermejo,[8,1] Aroosa Ijaz,[3,9,10] Supanut Thanasilp,[7,11] Eric R. Anschuetz,[12,13] and Zoë Holmes[7]

[1] *Information Sciences, Los Alamos National Laboratory, Los Alamos, NM 87545, USA*
[2] *Quantum Science Center, Oak Ridge, TN 37931, USA*
[3] *Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA*
[4] *Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*
[5] *Departamento de Física-IFLP/CONICET, Universidad Nacional de La Plata, C.C. 67, La Plata 1900, Argentina*
[6] *University of Strathclyde, Glasgow G1 1XQ, UK*
[7] *Institute of Physics, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne CH-1015, Switzerland*
[8] *Donostia International Physics Center, Paseo Manuel de Lardizabal 4, San Sebastián E-20018, Spain*
[9] *Department of Physics and Astronomy, University of Waterloo, Ontario, N2L 3G1, Canada*
[10] *Vector Institute, MaRS Centre, Toronto, Ontario, M5G 1M1, Canada*
[11] *Chula Intelligent and Complex Systems, Department of Physics, Faculty of Science, Chulalongkorn University, Bangkok, Thailand, 10330*
[12] *Institute for Quantum Information and Matter, Caltech, Pasadena 91125, USA*
[13] *Walter Burke Institute for Theoretical Physics, Caltech, Pasadena 91125, USA*

A large amount of effort has recently been put into understanding the barren plateau phenomenon. In this perspective article, we face the increasingly loud elephant in the room and ask a question that has been hinted at by many but not explicitly addressed: *Can the structure that allows one to avoid barren plateaus also be leveraged to efficiently simulate the loss classically?* We present strong evidence that commonly used models with provable absence of barren plateaus are also classically simulable, provided that one can collect some classical data from quantum devices during an initial data acquisition phase. This follows from the observation that barren plateaus result from a curse of dimensionality, and that current approaches for solving them end up encoding the problem into some small, classically simulable, subspaces. Thus, while stressing quantum computers can be essential for collecting data, our analysis sheds serious doubt on the non-classicality of the information processing capabilities of parametrized quantum circuits for barren plateau-free landscapes. We end by discussing caveats in our arguments, the role of smart initializations and the possibility of provably superpolynomial, or simply practical, advantages from running parametrized quantum circuits.

The Quantum Fourier Transform (again)

Inference, interference and invariance:
How the Quantum Fourier Transform can help to learn from data

David Wakeham and Maria Schuld
Xanadu, Toronto, ON, M5G 2C8, Canada

How can we take inspiration from a typical quantum algorithm to design heuristics for machine learning? A common blueprint, used from Deutsch-Josza to Shor's algorithm, is to place labeled information in superposition via an oracle, interfere in Fourier space, and measure. In this paper, we want to understand how this interference strategy can be used for *inference*, i.e. to generalize from finite data samples to a ground truth. Our investigative framework is built around the Hidden Subgroup Problem (HSP), which we transform into a learning task by replacing the oracle with classical training data. The standard quantum algorithm for solving the HSP uses the Quantum Fourier Transform to expose an invariant subspace, i.e., a subset of Hilbert space in which the hidden symmetry is manifest. Based on this insight, we propose an inference principle that "compares" the data to this invariant subspace, and suggest a concrete implementation via overlaps of quantum states. We hope that this leads to well-motivated quantum heuristics that can leverage symmetries for machine learning applications.

I. INTRODUCTION

The Hidden Subgroup Problem (HSP) [1, 2] is the task of discovering a subgroup from information about the way it partitions the parent group. While abstract, it neatly generalizes many problems solved by quantum algorithms, from Deutsch-Jozsa [3] to Simon's problem [4] to Shor's algorithms for period-finding and discrete logarithms [5]. The standard quantum routine for the HSP [6] has a common and embarrassingly simple blueprint: label all inputs, uniformly superpose, apply the Fourier transform, and measure. Formally, this samples from a

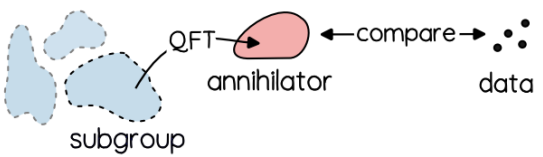


FIG. 1: Which hidden subgroup gave rise to the data? We propose to compare the data with the annihilator of a given subgroup. The annihilator is computed via a group Quantum Fourier Transform executed by a quantum computer.

Thank you for your attention!