So you want to build a quantum computer ...

Ben Criger [1]

[1]Qutech, TU Delft

February 27, 2018

READYING QUANTUM COMPUTING FOR LIFE'S BIGGEST MYSTERIES

December 1, 2017

Researchers at QuTech in the Netherlands and Intel are racing to engineer qubit crunching computer chips and specialized systems that use quantum mechanics. Focused on very small atoms and subatomic particles, this branch of physics may unlock the mysteries of nature, manage the complexities of digitally connected societies and maybe even solve some of the biggest problems facing humankind.

READYING QUANTUM COMPUTING FOR LIFE'S BIGGEST MYSTERIES

December 1, 2017

Researchers at QuTech in the Netherlands and Intel are racing to engineer qubit crunching computer chips and specialized systems that use quantum mechanics. Focused on very small atoms and subatomic particles, this branch of physics may unlock the mysteries of nature, manage the complexities of digitally connected societies and maybe even solve some of the biggest problems facing humankind.

READYING QUANTUM COMPUTING FOR LIFE'S BIGGEST MYSTERIES

December 1, 2017

He said quantum computing can lead to extremely fast computing, very sensitive detectors and the ability to create unbreakable encryption to protect important digital information.

Researchers at QuTech in the Netherlands and Intel are racing to engineer qubit crunching computer chips and specialized systems that use quantum mechanics. Focused on very small atoms and subatomic particles, this branch of physics may unlock the mysteries of nature, manage the complexities of digitally connected societies and maybe even solve some of the biggest problems facing humankind.

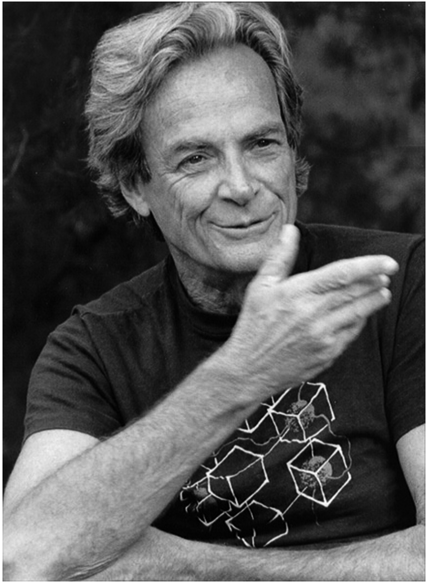
READYING QUANTUM COMPUTING FOR LIFE'S BIGGEST MYSTERIES

December 1, 2017

He said quantum computing can lead to extremely fast computing, very sensitive detectors and the ability to create unbreakable encryption to protect important digital information.

Other possible applications include drug design and materials science (modeling complex structures in nature like proteins) and logistics optimization (finding the most efficient of any number of possible travel routes).

You know the basics



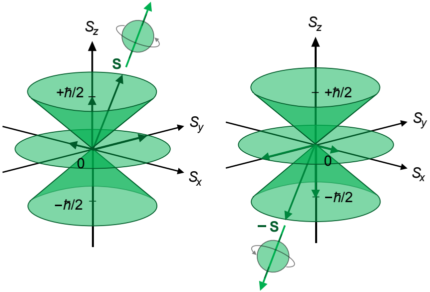
“Nature isn’t classical, dammit, and if you want to make a simulation of nature, you’d better make it quantum mechanical, and by golly it’s a wonderful problem, because it doesn’t look so easy.”

– Richard Feynman, 1982

You know the basics

What does this mean?

For simplicity, let's consider some *two-level systems* (or *qubits*):

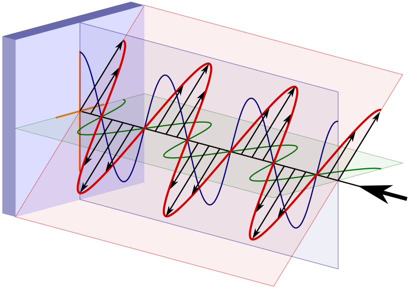


$|\uparrow\rangle \equiv \left|+\frac{1}{2}\right\rangle$ $\quad$ $|\downarrow\rangle \equiv \left|-\frac{1}{2}\right\rangle$

You know the basics

What does this mean?

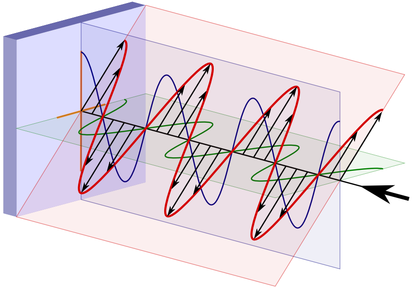
For simplicity, let's consider some *two-level systems* (or *qubits*):



You know the basics

What does this mean?

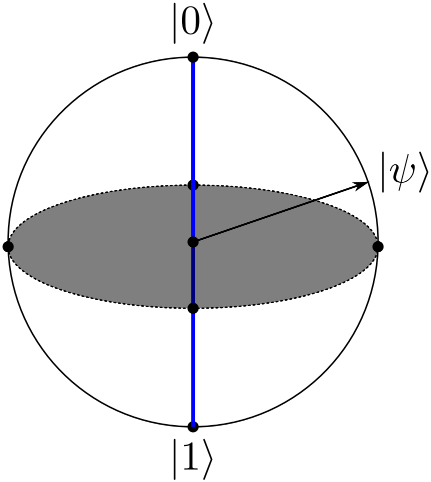
For simplicity, let's consider some *two-level systems* (or *qubits*):



States in these systems are *superpositions* of $|\uparrow\rangle/|\downarrow\rangle$ or $|\text{horizontal}\rangle/|\text{vertical}\rangle$, which we can denote $\alpha|0\rangle + \beta|1\rangle$ or $\begin{bmatrix}\alpha\\ \beta\end{bmatrix}$.

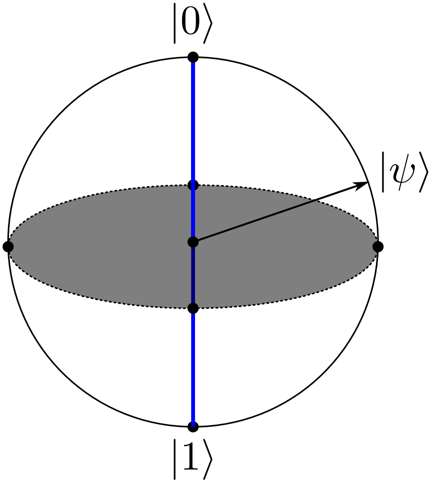
Superposition and its Implications

Superpositions have to be *normalised* ($|\alpha|^2 + |\beta|^2 = 1$) $\therefore \alpha = \cos(\theta)$ and $\beta = e^{i\phi}\sin(\theta)$



Superposition and its Implications

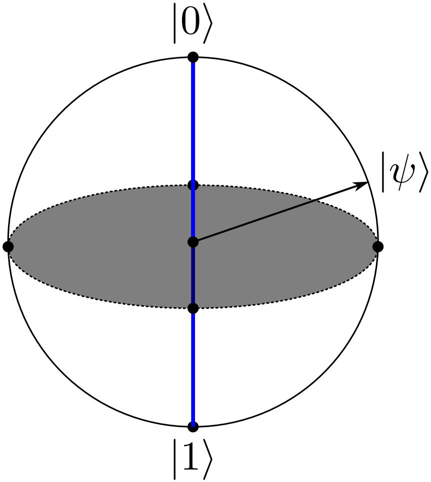
Superpositions have to be *normalised* ($|\alpha|^2 + |\beta|^2 = 1$) $\therefore \alpha = \cos(\theta)$ and $\beta = e^{i\phi}\sin(\theta)$



- *basis states* at the north/south poles

Superposition and its Implications

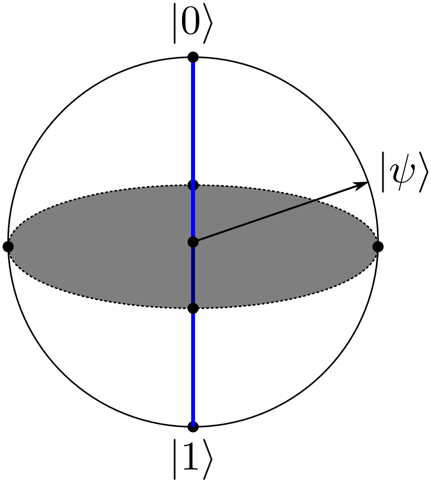
Superpositions have to be *normalised* $(|\alpha|^2 + |\beta|^2 = 1) \therefore \alpha = \cos(\theta)$ and $\beta = e^{i\phi}\sin(\theta)$



- *basis states* at the north/south poles
- *superpositions* on the surface of the sphere

Superposition and its Implications

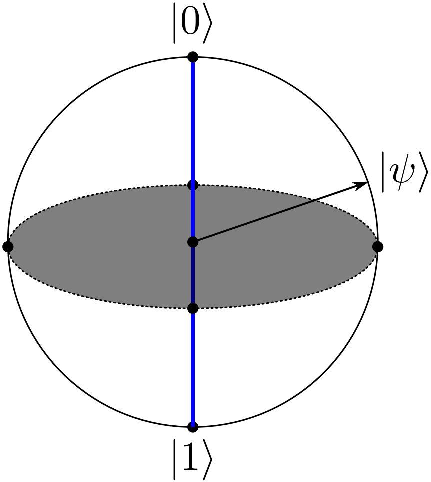
Superpositions have to be *normalised* $(|\alpha|^2 + |\beta|^2 = 1) \therefore \alpha = \cos(\theta)$ and $\beta = e^{i\phi}\sin(\theta)$



- *basis states* at the north/south poles
- *superpositions* on the surface of the sphere
- *probabilistic mixtures* on the interior of the sphere (ask me about this later!)

Superposition and its Implications

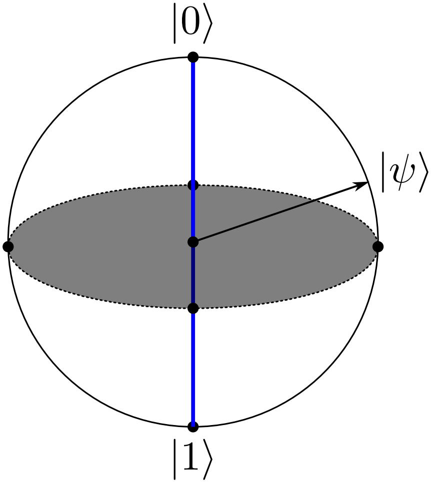
Superpositions have to be *normalised* ($|\alpha|^2 + |\beta|^2 = 1$) $\therefore$ $\alpha = \cos(\theta)$ and $\beta = e^{i\phi}\sin(\theta)$



- *basis states* at the north/south poles
- *superpositions* on the surface of the sphere
- *probabilistic mixtures* on the interior of the sphere (ask me about this later!)
- *classical states* on the blue axis

Superposition and its Implications

Superpositions have to be *normalised* $(|\alpha|^2 + |\beta|^2 = 1)$ $\therefore \alpha = \cos(\theta)$ and $\beta = e^{i\phi}\sin(\theta)$



- *basis states* at the north/south poles
- *superpositions* on the surface of the sphere
- *probabilistic mixtures* on the interior of the sphere (ask me about this later!)
- *classical states* on the blue axis
- *operations on single qubits* given by rotations on the sphere

Superposition and its Implications

States on multiple qubits are superpositions of basis states:

$$|00\cdots00\rangle\,, |00\cdots01\rangle\,, \ldots, |11\cdots10\rangle\,, |11\cdots11\rangle$$

Superposition and its Implications

States on multiple qubits are superpositions of basis states:

$$|00\cdots00\rangle, |00\cdots01\rangle, \ldots, |11\cdots10\rangle, |11\cdots11\rangle$$

In general, any superposition of all 2^n terms can occur, so we'd need $O(2^n)$ floating point numbers just to store a single quantum state.

Superposition and its Implications

States on multiple qubits are superpositions of basis states:

$$|00\cdots00\rangle, |00\cdots01\rangle, \ldots, |11\cdots10\rangle, |11\cdots11\rangle$$

In general, any superposition of all 2^n terms can occur, so we'd need $O(2^n)$ floating point numbers just to store a single quantum state.

Classical computers are exponentially inefficient at simulating quantum mechanics in general.

Superposition and its Implications

In *circuit-based* quantum computing[1], we typically assume that we can:

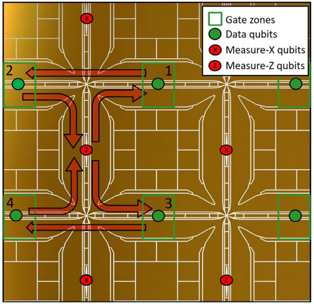
[1] Look up "DQC1" and "adiabatic quantum computation" for different types of quantum computing.

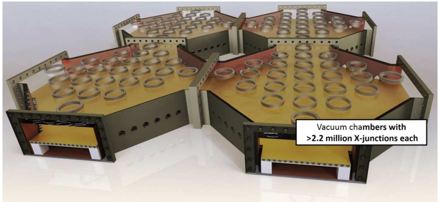
Superposition and its Implications

In *circuit-based* quantum computing[1], we typically assume that we can:

Scale the System Up

The n^{th} qubit shouldn't be much harder to build than the first.



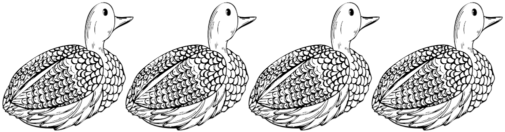


[1] Look up "DQC1" and "adiabatic quantum computation" for different types of quantum computing.

Superposition and its Implications

In *circuit-based* quantum computing[1], we typically assume that we can:

Prepare Initial States

$$|00\cdots 0\rangle =$$ 

Garbage in, garbage out.

[1] Look up "DQC1" and "adiabatic quantum computation" for different types of quantum computing.

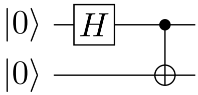
Superposition and its Implications

In *circuit-based* quantum computing[1], we typically assume that we can:

Perform Single- and Two-Qubit Gates

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

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



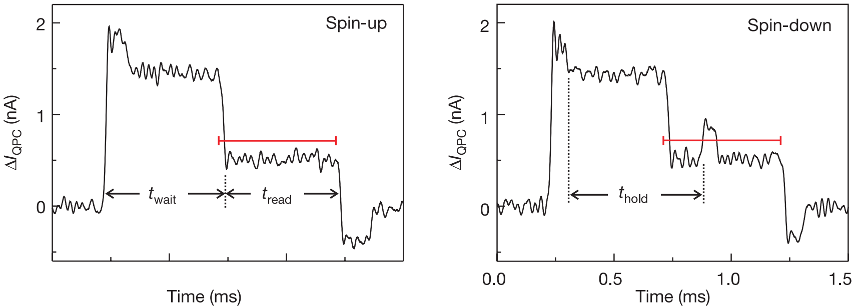
$$|00\rangle \underset{H_1}{\longmapsto} \frac{|00\rangle + |10\rangle}{\sqrt{2}} \underset{\text{CNOT}}{\longmapsto} \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

[1] Look up “DQC1” and “adiabatic quantum computation” for different types of quantum computing.

In *circuit-based* quantum computing[1], we typically assume that we can:

Read Out States

Making classical signals out of quantum states



[1] Look up "DQC1" and "adiabatic quantum computation" for different types of quantum computing.

Superposition and its Implications

This gives us the power to implement algorithms *in theory*:

Algorithm: Quantum Simulation
Speedup: Superpolynomial
Description: It is believed that for any physically realistic Hamiltonian H on n degrees of freedom, the corresponding time evolution operator e^{-iHt} can be implemented using poly(n,t) gates. Unless BPP=BQP, this problem is not solvable in general on a classical computer in polynomial time. Many techniques for quantum simulation have been developed for general classes of Hamiltonians [25,95,92,5,12,170,205,211,244,245,278,293,294,295,372,382], chemical dynamics [63,68,227,310,375], condensed matter physics [1,99, 145], relativistic quantum mechanics (the Dirac and Klein-Gordon equations) [367,369,370,371], open quantum systems [376, 377,378,379], and quantum field theory [107,166,228,229,230,368]. The exponential complexity of classically simulating quantum systems led Feynman to first propose that quantum computers might outperform classical computers on certain tasks [40]. Although the problem of finding ground energies of local Hamiltonians is QMA-complete and therefore probably requires exponential time on a quantum computer in the worst case, quantum algorithms have been developed to approximate ground [102,231,232,233,234,235,308,321,322,380,381] and thermal [132,121,281,282,307] states for some classes of Hamiltonians. Efficient quantum algorithms have been also obtained for preparing certain classes of tensor network states [323,324,325,326,327,328].

From `https://math.nist.gov/quantum/zoo/`.

Superposition and its Implications

This gives us the power to implement algorithms *in theory*:

Algorithm: Pattern matching
Speedup: Superpolynomial
Description: Given strings *T* of length *n* and *P* of length *m* < *n*, both from some finite alphabet, the pattern matching problem is to find an occurrence of *P* as a substring of *T* or to report that *P* is not a substring of *T*. More generally, *T* and *P* could be *d*-dimensional arrays rather than one-dimensional arrays (strings). Then, the pattern matching problem is to return the location of *P* as a $m \times m \times \ldots \times m$ block within the $n \times n \times \ldots \times n$ array *T* or report that no such location exists. The $\Omega(\sqrt{N})$ query lower bound for unstructured search [216] implies that the worst-case quantum query complexity of this problem is $\Omega(\sqrt{n} + \sqrt{m})$. A quantum algorithm achieving this, up to logarithmic factors, was obtained in [217]. This quantum algorithm works through the use of Grover's algorithm together with a classical method called deterministic sampling. More recently, Montanaro showed that superpolynomial quantum speedup can be achieved on average case instances of pattern matching, provided that *m* is greater than logarithmic in *n*. Specifically, the quantum algorithm given in [215] solves average case pattern matching in $\widetilde{O}((n/m)^{d/2} 2^{O(d^{3/2}\sqrt{\log m})})$ time. This quantum algorithm is constructed by generalizing Kuperberg's quantum sieve algorithm [66] for dihedral hidden subgroup and hidden shift problems so that it can operate in *d* dimensions and accomodate small amounts of noise, and then classically reducing the pattern matching problem to this noisy *d*-dimensional version of hidden shift.

From `https://math.nist.gov/quantum/zoo/`.

Difficulties in Quantum Computing

What about implementing algorithms *in practice*?

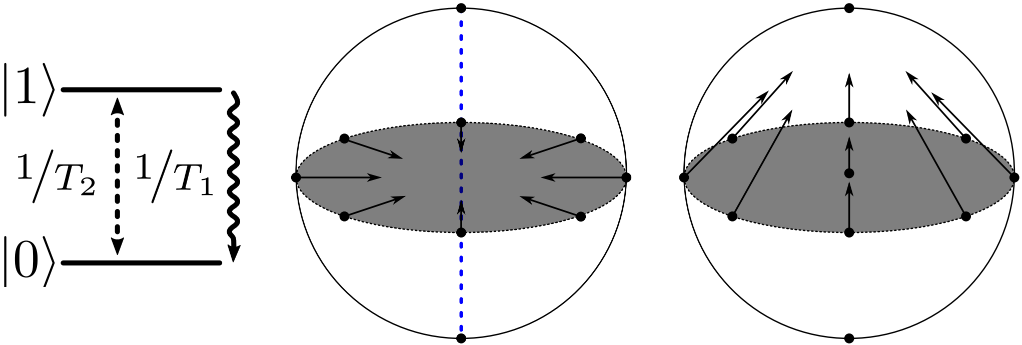
There are three main obstacles:

Difficulties in Quantum Computing

What about implementing algorithms *in practice*?

There are three main obstacles:

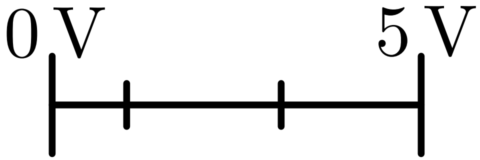
Decoherence

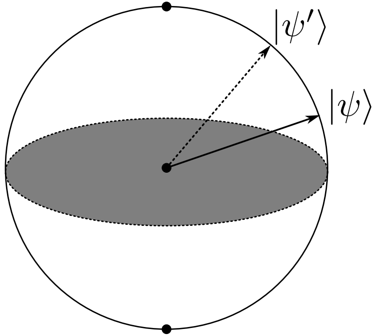


What about implementing algorithms *in practice*?

There are three main obstacles:

Analog Control





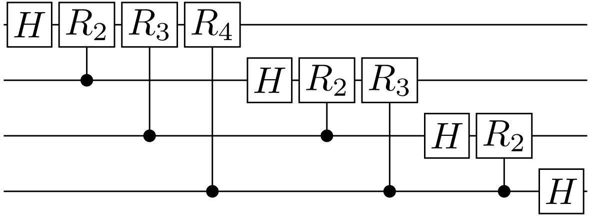
Difficulties in Quantum Computing

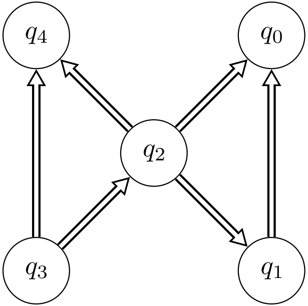
What about implementing algorithms *in practice*?

There are three main obstacles:

Limited Connectivity

Not all pairs of qubits can participate in a CNOT directly





Questions & Answers

Can we *simulate* the devices that theorists imagine using the noisy ones from the lab?

Questions & Answers

Can we *simulate* the devices that theorists imagine using the noisy ones from the lab?

Yes:

Questions & Answers

Can we *simulate* the devices that theorists imagine using the noisy ones from the lab?

Yes:

Quantum Error Correction defines sets of *states* where the effects of noise can be reversed in theory.

Questions & Answers

Can we *simulate* the devices that theorists imagine using the noisy ones from the lab?

Yes:

Quantum Error Correction defines sets of *states* where the effects of noise can be reversed in theory.

Quantum Fault Tolerance defines sets of *computations* where the effects of noise can be reversed in theory.

Questions & Answers

Can we *simulate* the devices that theorists imagine using the noisy ones from the lab?

Yes:

Quantum Error Correction defines sets of *states* where the effects of noise can be reversed in theory.

Quantum Fault Tolerance defines sets of *computations* where the effects of noise can be reversed in theory.

So far, we think we can implement quantum computing with *logarithmic overhead* in a *planar* quantum computer

Questions & Answers

Can we *simulate* the devices that theorists imagine using the noisy ones from the lab?
Yes:

Quantum Error Correction defines sets of *states* where the effects of noise can be reversed in theory.

Quantum Fault Tolerance defines sets of *computations* where the effects of noise can be reversed in theory.

$$O\left(N^f \log(\text{poly}(N))\right) < O(N^s) \text{ if } f < s$$

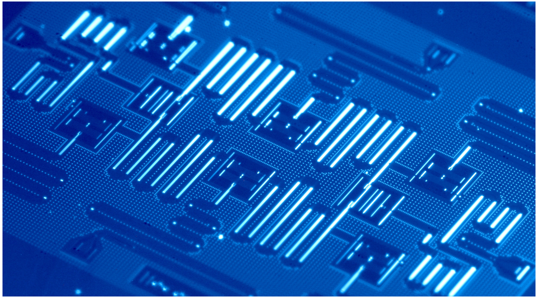
Questions & Answers

Can we *simulate* the devices that theorists imagine using the noisy ones from the lab?

Yes:

Quantum Error Correction defines sets of *states* where the effects of noise can be reversed in theory.

Quantum Fault Tolerance defines sets of *computations* where the effects of noise can be reversed in theory.



Quantum Error Correction

Quantum Error Correction

- The first 'baby step' toward implementing quantum computing

Quantum Error Correction

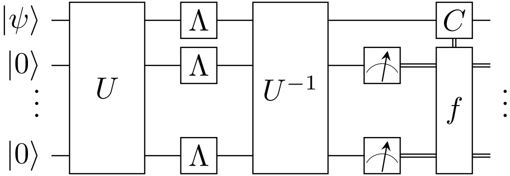
- The first ‘baby step’ toward implementing quantum computing
- Most operations (physical states, unitaries) are perfect

Quantum Error Correction

- The first ‘baby step’ toward implementing quantum computing
- Most operations (physical states, unitaries) are perfect
- Errors only occur when we send a state through a *channel*:

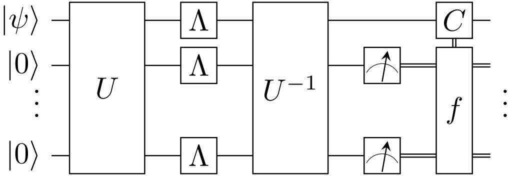
Quantum Error Correction

- The first 'baby step' toward implementing quantum computing
- Most operations (physical states, unitaries) are perfect
- Errors only occur when we send a state through a *channel*:



Quantum Error Correction

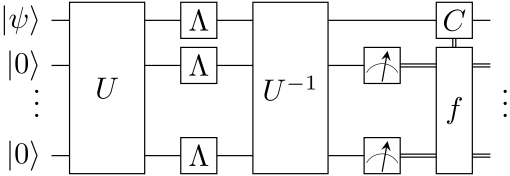
- The first 'baby step' toward implementing quantum computing
- Most operations (physical states, unitaries) are perfect
- Errors only occur when we send a state through a *channel*:



- The states $U|00\cdots0\rangle$ and $U|10\cdots0\rangle$ that span our space of messages are called *code states*

Quantum Error Correction

- The first 'baby step' toward implementing quantum computing
- Most operations (physical states, unitaries) are perfect
- Errors only occur when we send a state through a *channel*:



- The states $U|00\cdots0\rangle$ and $U|10\cdots0\rangle$ that span our space of messages are called *code states*
- We usually define these code states to be the *joint* $+1$ *eigenspace* of some operators called *stabilisers*

Aside: Some Brief Examples of Stabilisers

The state $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$ that we saw earlier is a *stabiliser state*:

Aside: Some Brief Examples of Stabilisers

The state $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$ that we saw earlier is a *stabiliser state*:

$$X : X\,|0\rangle = |1\rangle\,,\ \ X\,|1\rangle = |0\rangle \qquad\qquad Z : Z\,|0\rangle = |0\rangle\,,\ \ Z\,|1\rangle = -\,|1\rangle$$

Aside: Some Brief Examples of Stabilisers

The state $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$ that we saw earlier is a *stabiliser state*:

$$X : X\,|0\rangle = |1\rangle\,,\;\; X\,|1\rangle = |0\rangle \qquad\qquad Z : Z\,|0\rangle = |0\rangle\,,\;\; Z\,|1\rangle = -\,|1\rangle$$

$$X \otimes X\,\frac{|00\rangle + |11\rangle}{\sqrt{2}} = \frac{|11\rangle + |00\rangle}{\sqrt{2}} = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

Aside: Some Brief Examples of Stabilisers

The state $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$ that we saw earlier is a *stabiliser state*:

$$X : X|0\rangle = |1\rangle\,,\ X|1\rangle = |0\rangle \qquad Z : Z|0\rangle = |0\rangle\,,\ Z|1\rangle = -|1\rangle$$

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

$$Z \otimes Z \frac{|00\rangle + |11\rangle}{\sqrt{2}} = \frac{|00\rangle + (-1)(-1)|11\rangle}{\sqrt{2}} = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

Aside: Some Brief Examples of Stabilisers

The state $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$ that we saw earlier is a *stabiliser state*:

$$X : X\,|0\rangle = |1\rangle\,,\;\; X\,|1\rangle = |0\rangle \qquad\qquad Z : Z\,|0\rangle = |0\rangle\,,\;\; Z\,|1\rangle = -\,|1\rangle$$

$$X \otimes X\,\frac{|00\rangle + |11\rangle}{\sqrt{2}} = \frac{|11\rangle + |00\rangle}{\sqrt{2}} = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

$$Z \otimes Z\,\frac{|00\rangle + |11\rangle}{\sqrt{2}} = \frac{|00\rangle + (-1)(-1)\,|11\rangle}{\sqrt{2}} = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

- These operators are *observables*, so we can measure them to determine if the state is in the code space.

Aside: Some Brief Examples of Stabilisers

The state $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$ that we saw earlier is a *stabiliser state*:

$$X : X\,|0\rangle = |1\rangle\,, \;\; X\,|1\rangle = |0\rangle \qquad\qquad Z : Z\,|0\rangle = |0\rangle\,, \;\; Z\,|1\rangle = -\,|1\rangle$$

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

$$Z \otimes Z \frac{|00\rangle + |11\rangle}{\sqrt{2}} = \frac{|00\rangle + (-1)(-1)\,|11\rangle}{\sqrt{2}} = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

- These operators are *observables*, so we can measure them to determine if the state is in the code space.
- We call the measurement results a *syndrome*, since they're a list of 'symptoms'.

Aside: Some Brief Examples of Stabilisers

The state $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$ that we saw earlier is a *stabiliser state*:

$$X : X|0\rangle = |1\rangle\,,\ X|1\rangle = |0\rangle \qquad Z : Z|0\rangle = |0\rangle\,,\ Z|1\rangle = -|1\rangle$$

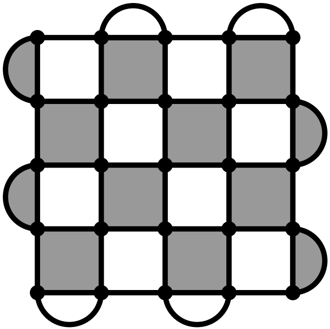
$$X \otimes X \frac{|00\rangle + |11\rangle}{\sqrt{2}} = \frac{|11\rangle + |00\rangle}{\sqrt{2}} = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

$$Z \otimes Z \frac{|00\rangle + |11\rangle}{\sqrt{2}} = \frac{|00\rangle + (-1)(-1)|11\rangle}{\sqrt{2}} = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

- These operators are *observables*, so we can measure them to determine if the state is in the code space.
- We call the measurement results a *syndrome*, since they're a list of 'symptoms'.
- Stabiliser states can be mapped to other stabiliser states by H, CNOT and P, and we can simulate them efficiently on classical computers.

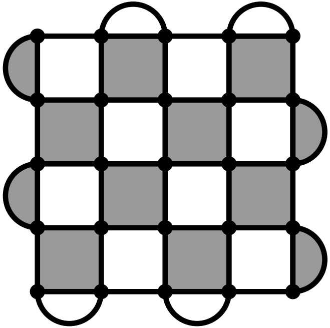
The Surface Code

To encode states for transmission, we mostly use this planar code:



The Surface Code

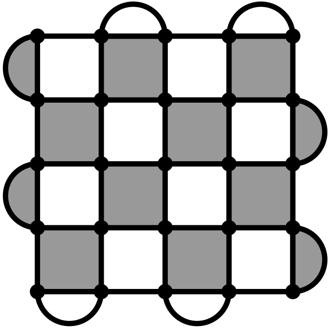
To encode states for transmission, we mostly use this planar code:



- Qubits are at the corners of a square lattice

The Surface Code

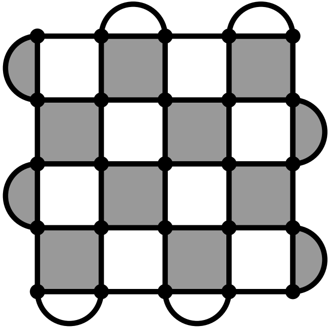
To encode states for transmission, we mostly use this planar code:



- Qubits are at the corners of a square lattice
- **White** stabilisers of the form $X^{\otimes 2}$, $X^{\otimes 4}$

The Surface Code

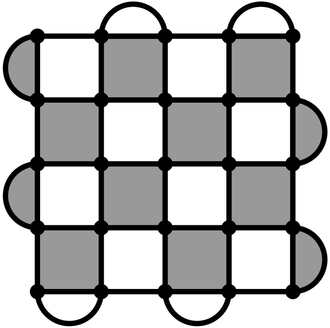
To encode states for transmission, we mostly use this planar code:



- Qubits are at the corners of a square lattice
- **White** stabilisers of the form $X^{\otimes 2}$, $X^{\otimes 4}$
- **Grey** stabilisers of the form $Z^{\otimes 2}$, $Z^{\otimes 4}$

The Surface Code

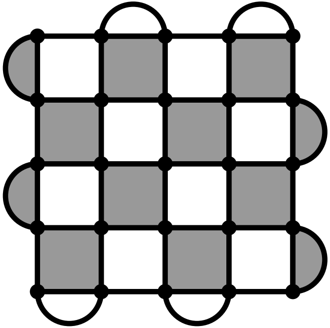
To encode states for transmission, we mostly use this planar code:



- Qubits are at the corners of a square lattice
- **White** stabilisers of the form $X^{\otimes 2}$, $X^{\otimes 4}$
- **Grey** stabilisers of the form $Z^{\otimes 2}$, $Z^{\otimes 4}$
- Logical error rates scale as $c(L)p^{\lfloor L/2 \rfloor}$

The Surface Code

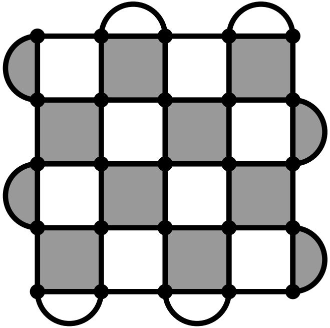
To encode states for transmission, we mostly use this planar code:



- Qubits are at the corners of a square lattice
- **White** stabilisers of the form $X^{\otimes 2}$, $X^{\otimes 4}$
- **Grey** stabilisers of the form $Z^{\otimes 2}$, $Z^{\otimes 4}$
- Logical error rates scale as $c(L)p^{\lfloor L/2 \rfloor}$
- Beneath the *threshold error rate*, bigger is better

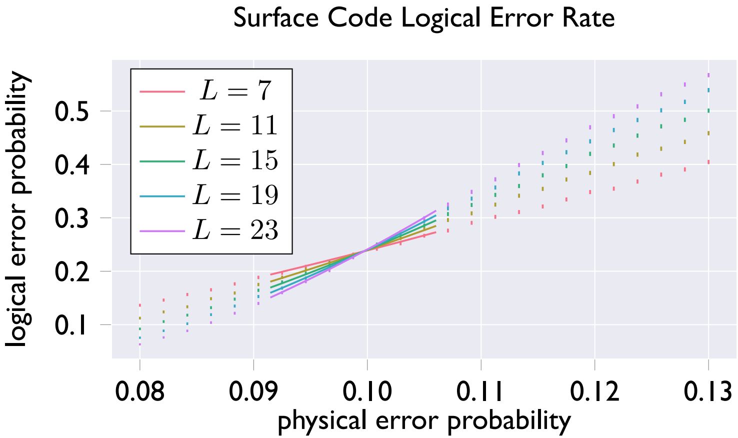
The Surface Code

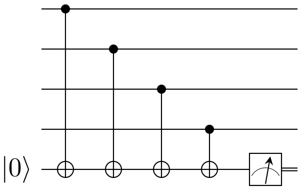
To encode states for transmission, we mostly use this planar code:



- Qubits are at the corners of a square lattice
- **White** stabilisers of the form $X^{\otimes 2}$, $X^{\otimes 4}$
- **Grey** stabilisers of the form $Z^{\otimes 2}$, $Z^{\otimes 4}$
- Logical error rates scale as $c(L)p^{\lfloor L/2 \rfloor}$
- Beneath the *threshold error rate*, bigger is better
- We usually measure the stabilisers using *ancilla qubits* and small circuits

The Surface Code



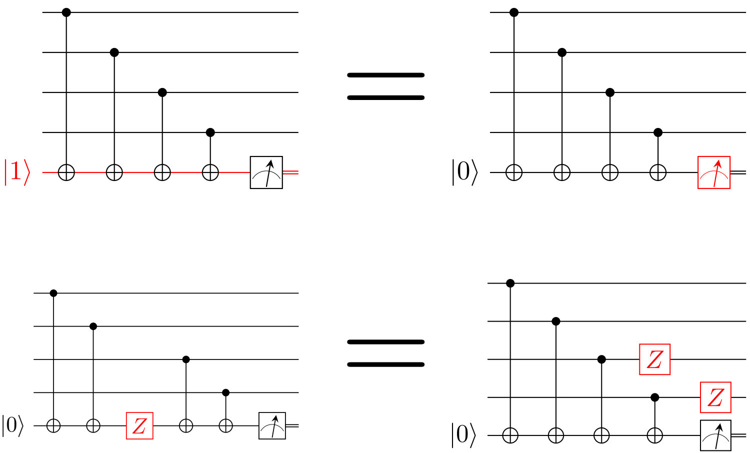


Fault Tolerance in Quantum Memories

Having a code is good, but not enough to protect against faulty operations:

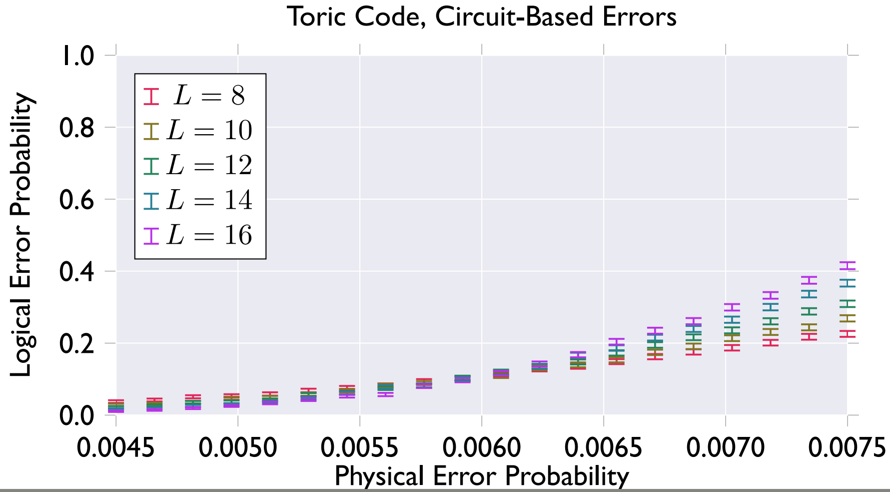
Fault Tolerance in Quantum Memories

Having a code is good, but not enough to protect against faulty operations:



Fault Tolerance in Quantum Memories

By repeating measurements and comparing the results, we can determine which operations are faulty and correct the effect on the data qubits:



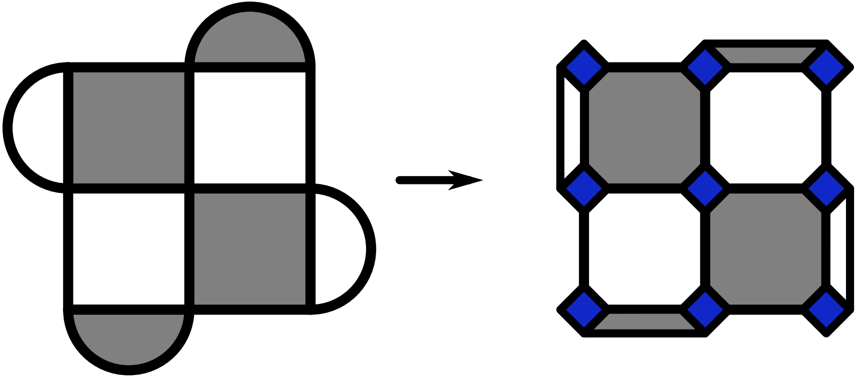
My Research

Codes Specialised to Hardware – arXiv:1604.04062v2

Different implementations of two-qubit gates can work better or worse, and we can design codes to take advantage:

Codes Specialised to Hardware – arXiv:1604.04062v2

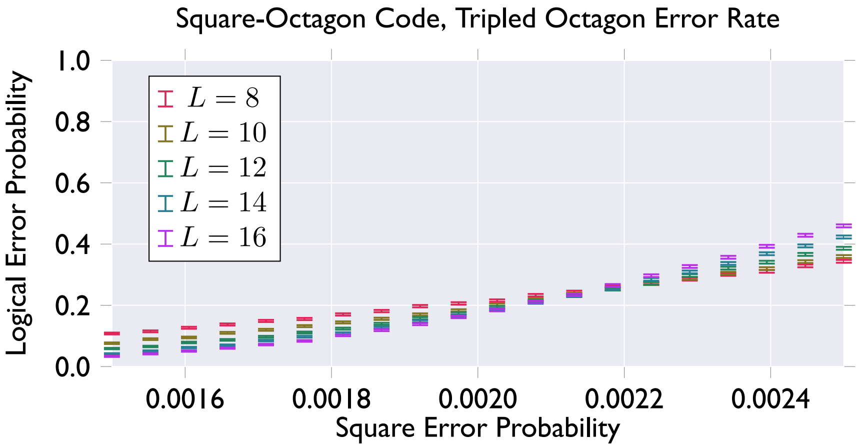
Different implementations of two-qubit gates can work better or worse, and we can design codes to take advantage:



My Research

Codes Specialised to Hardware – arXiv:1604.04062v2

Different implementations of two-qubit gates can work better or worse, and we can design codes to take advantage:



My Research

Better Inference Algorithms – arXiv:1709.02154

My Research

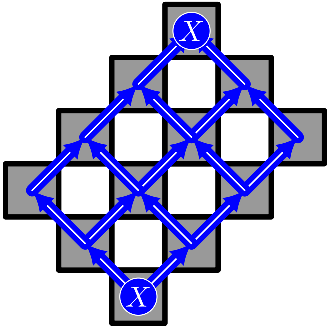
Better Inference Algorithms – arXiv:1709.02154

- It's not trivial to determine which correction to apply, given a syndrome

My Research

Better Inference Algorithms – arXiv:1709.02154

- It's not trivial to determine which correction to apply, given a syndrome
- Better algorithms for this task lead to better threshold error rates:

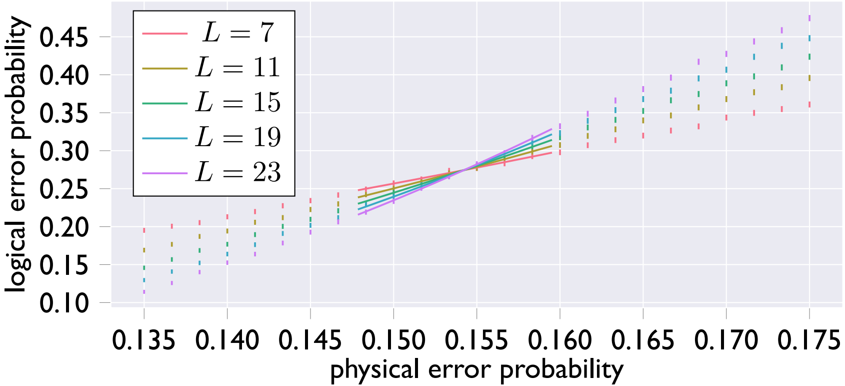


My Research

Better Inference Algorithms – arXiv:1709.02154

- It's not trivial to determine which correction to apply, given a syndrome
- Better algorithms for this task lead to better threshold error rates:

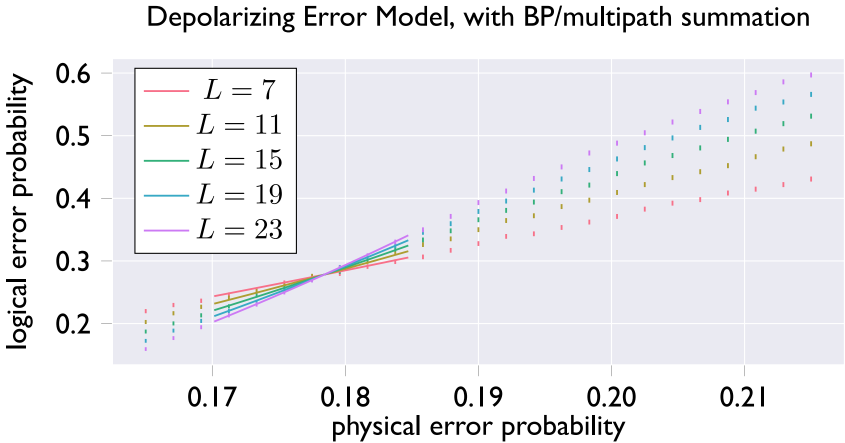
Depolarizing Error Model, without BP/multipath summation



My Research

Better Inference Algorithms – arXiv:1709.02154

- It's not trivial to determine which correction to apply, given a syndrome
- Better algorithms for this task lead to better threshold error rates:

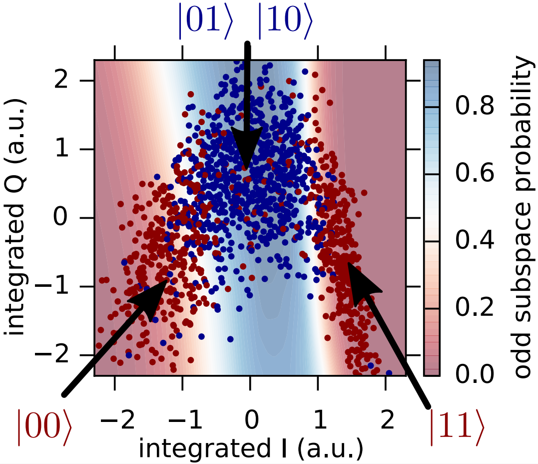


My Research

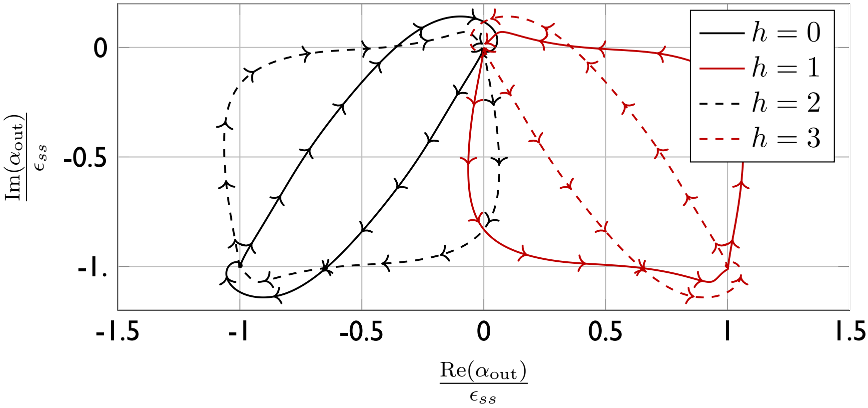
Direct Quantum Measurements – arXiv:1510.03211 & arXiv:1712.06141

We can couple multiple qubits to a measurement apparatus at once, measuring high-weight observables directly:

We can couple multiple qubits to a measurement apparatus at once, measuring high-weight observables directly:



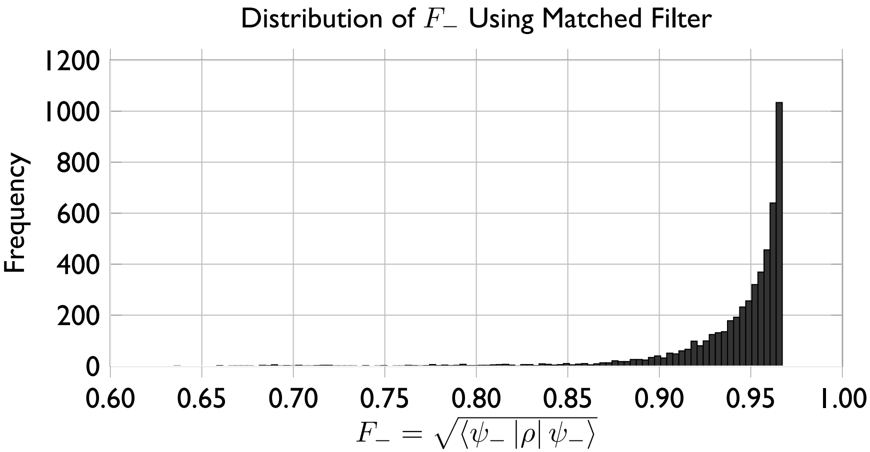
We can couple multiple qubits to a measurement apparatus at once, measuring high-weight observables directly:



My Research

Direct Quantum Measurements – arXiv:1510.03211 & arXiv:1712.06141

We can couple multiple qubits to a measurement apparatus at once, measuring high-weight observables directly:



Conclusions

Conclusions

- There are plenty of good reasons to build a quantum computer

Conclusions

- There are plenty of good reasons to build a quantum computer
- There are also plenty of obstacles to building a quantum computer

Conclusions

- There are plenty of good reasons to build a quantum computer
- There are also plenty of obstacles to building a quantum computer
- It’s not obviously impossible, though

Conclusions

- There are plenty of good reasons to build a quantum computer
- There are also plenty of obstacles to building a quantum computer
- It's not obviously impossible, though
- The more we know about the real capabilities of the hardware, the better we'll be able to do

To Learn More

- QuTech: `qutech.nl` & `qutechacademy.nl`
- Quantum Error Correction & Fault Tolerance: `pirsa.org/C17045`
- These slides: `https://github.com/bcriger/slide_decks`
- My research: `scirate.com/ben-criger/papers`
- My Twitter: `@bcriger`