Verification of NISQ Devices

•••

From Benchmarking to Protocol Verification
QuHackEd 2019

What are NISQ Devices?

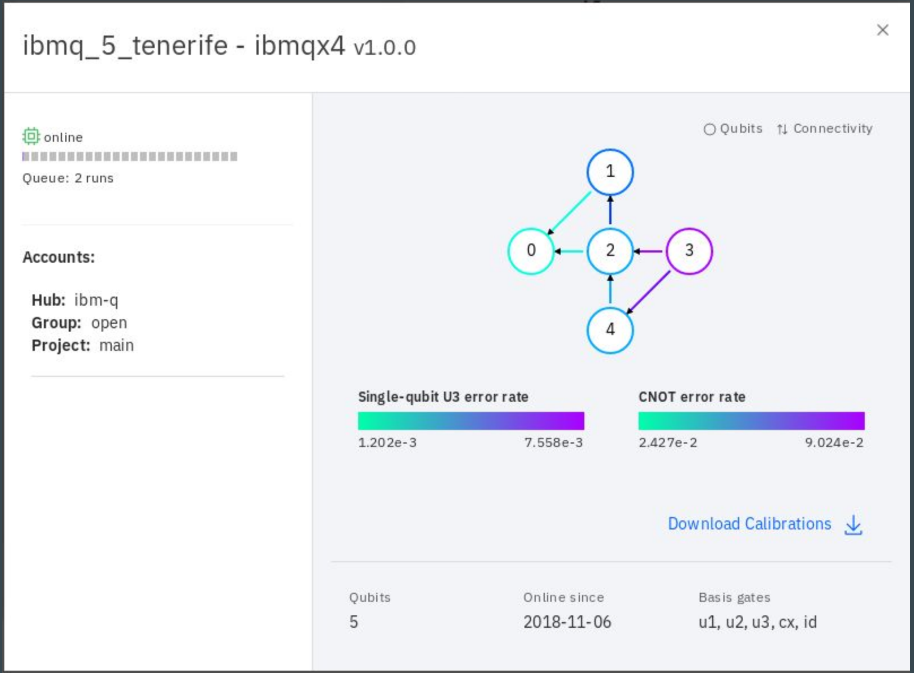
Noisy Intermediate Scale Quantum Devices

- Few qubits: 100-200
- Limited architecture
    - Cannot always directly connect all qubits
- Lots of Noise (I mean really... wow)
- No fault tolerance
    - Error correction requires significantly more qubits

Noisy Intermediate Scale Quantum Devices

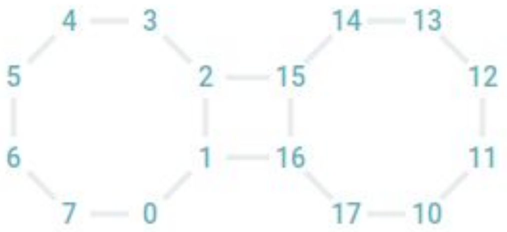
For example...

IBM-Q 5 Tenerife



IBM-Q 16 Melbourne

Rigetti Aspen-4-16Q-A



16 QUBITS

Aspen-4-16Q-A

T1	25.24 μs	fRO	93.43%
T2	19.89 μs	fCZ	90.81% ± 0.24%
f1QRB	82.19%	fBellState	91.08%
fActiveReset	95.41%		

Qubit Counter

www.qubitcounter.com

What Would We Like From Verification?

In An Ideal World

- Is the computation being performed on the quantum computer the one I want?
- Is the state my quantum computer is preparing the one I wanted?

In The Real World

- If you want to check that ANY computation is being performed correctly, you need A LOT of qubits
- To be totally sure, you might need a small quantum computer of your own

What Can We Expect From Verification?

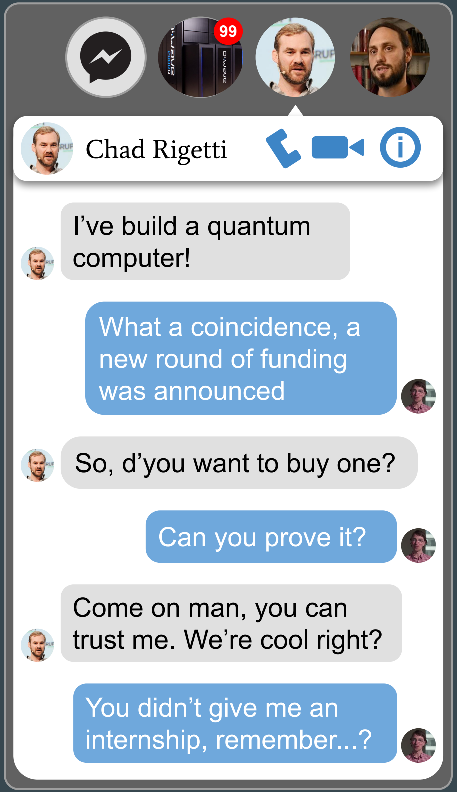
You Might Just About Get…

- Is my device doing anything quantum at all?
- Is the noise level reasonable?
- Is the distribution of outputs close to what it was meant to be?

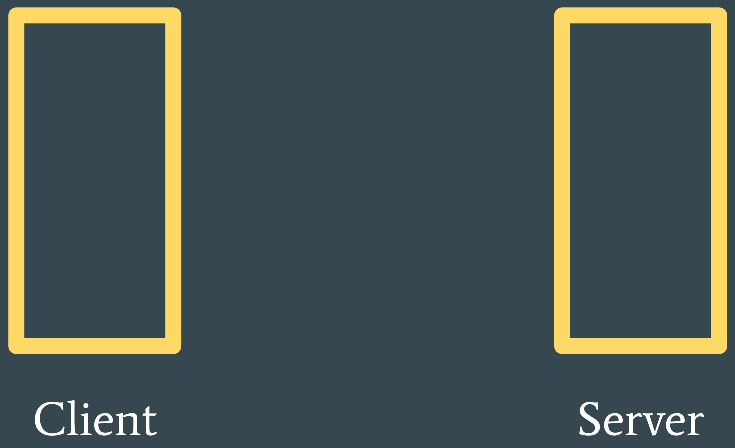
Hypothesis Testing

Is my device doing anything quantum at all?

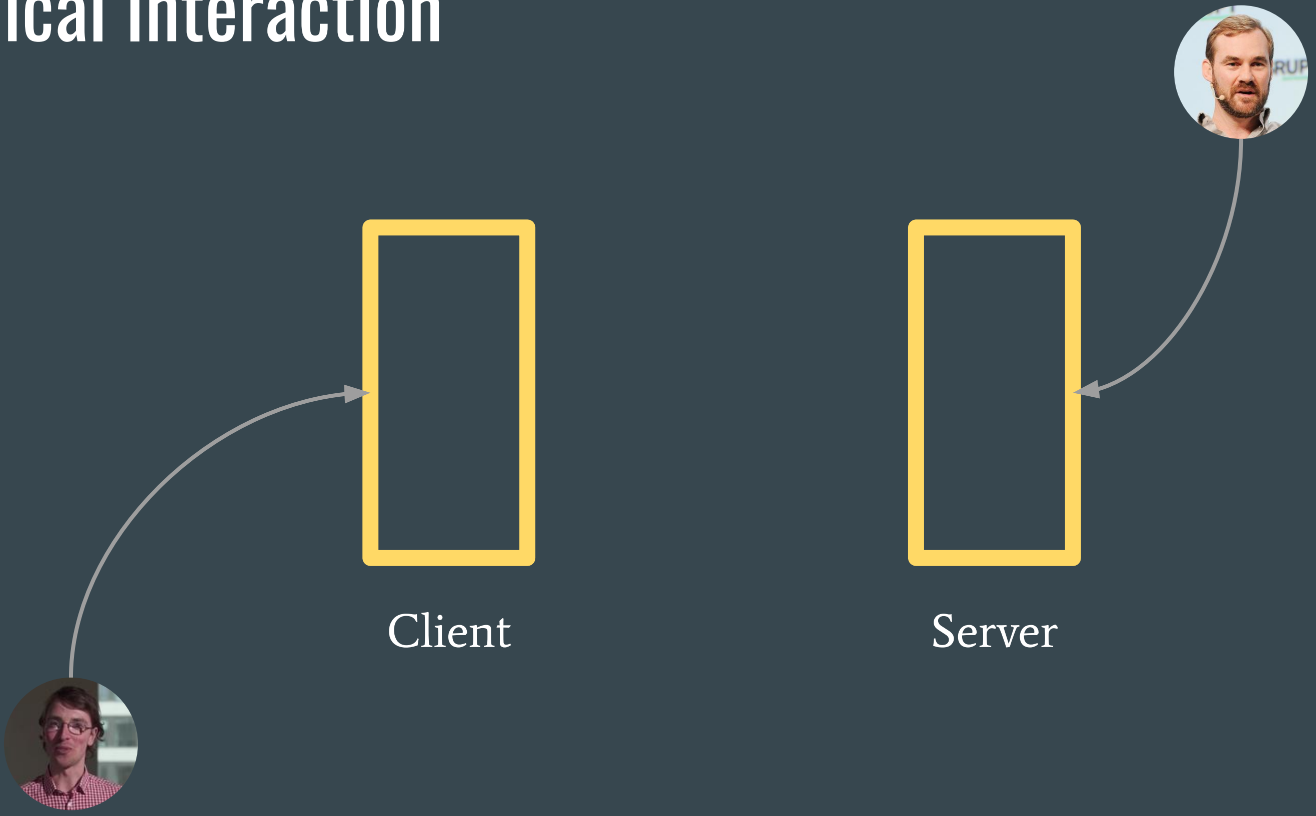
The Setting


99
Chad Rigetti
I've build a quantum computer!
What a coincidence, a new round of funding was announced
So, d'you want to buy one?
Can you prove it?
Come on man, you can trust me. We're cool right?
You didn't give me an internship, remember...?

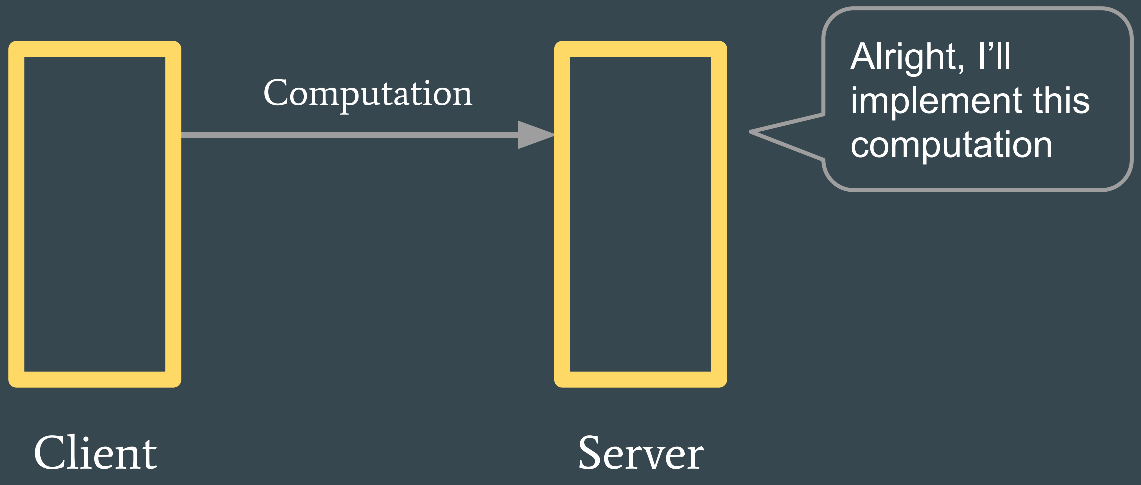
A Typical Interaction



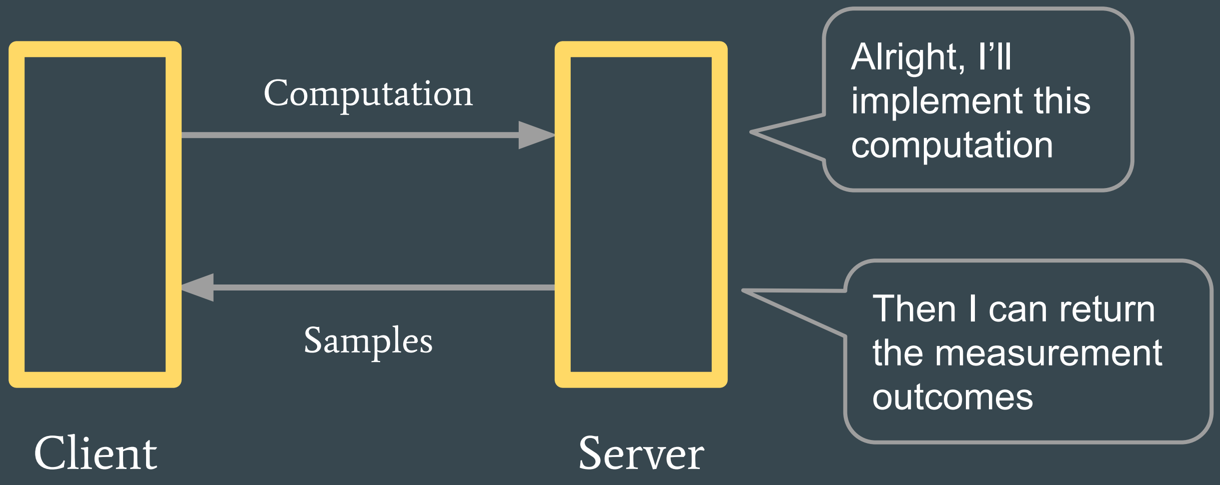
A Typical Interaction



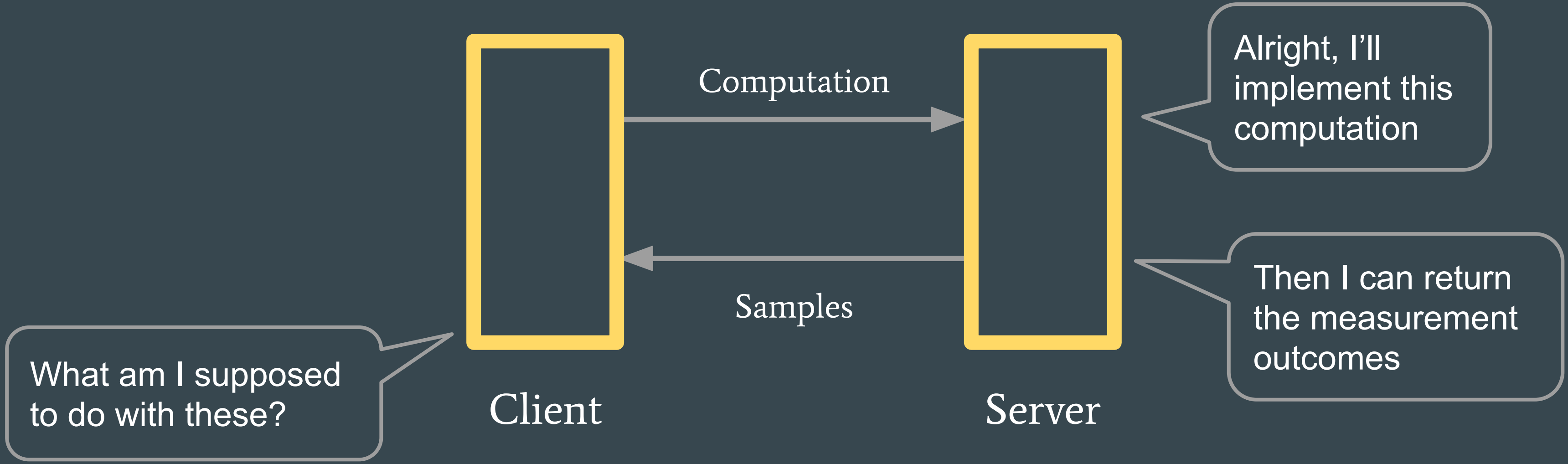
A Typical Interaction



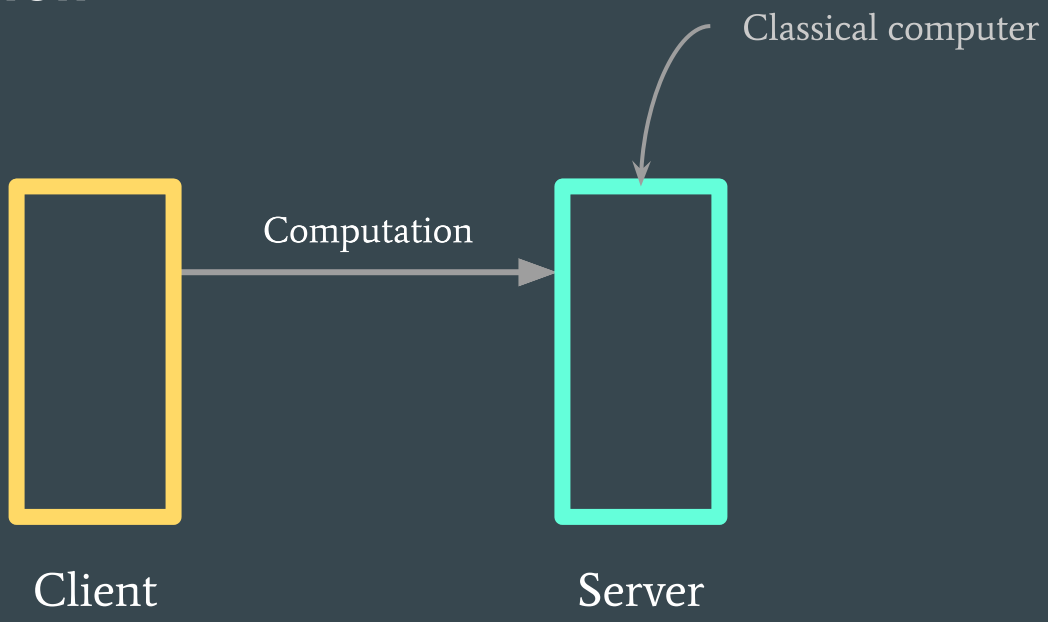
A Typical Interaction



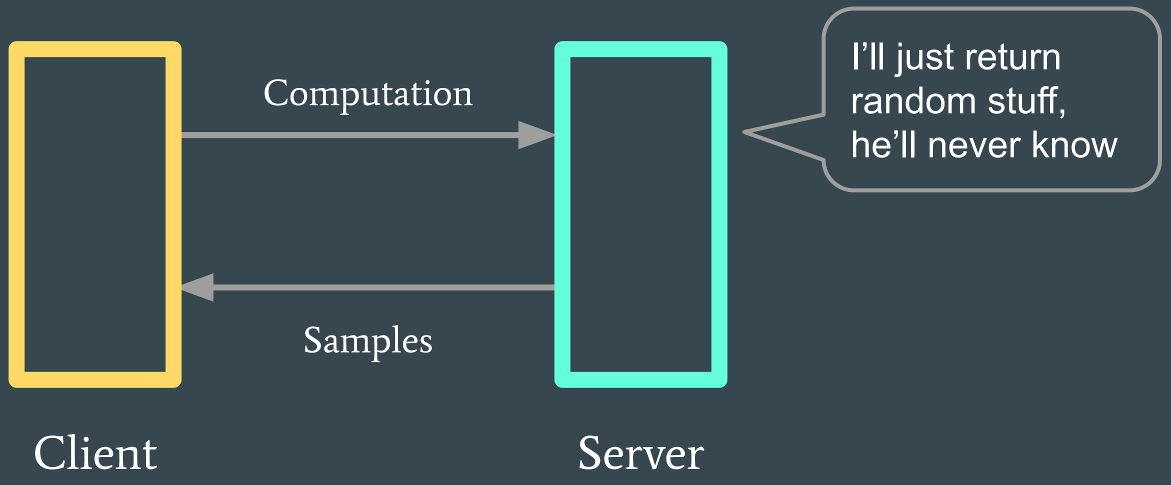
A Typical Interaction



A Typical Interaction



A Typical Interaction



Superiority Null Hypothesis

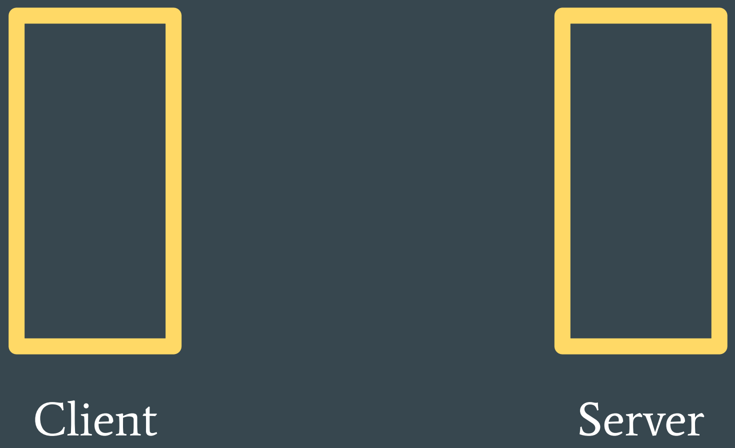
The set of samples which I have in my possession were drawn from a distribution produced by a classical computer in polynomial time

Superiority Null Hypothesis

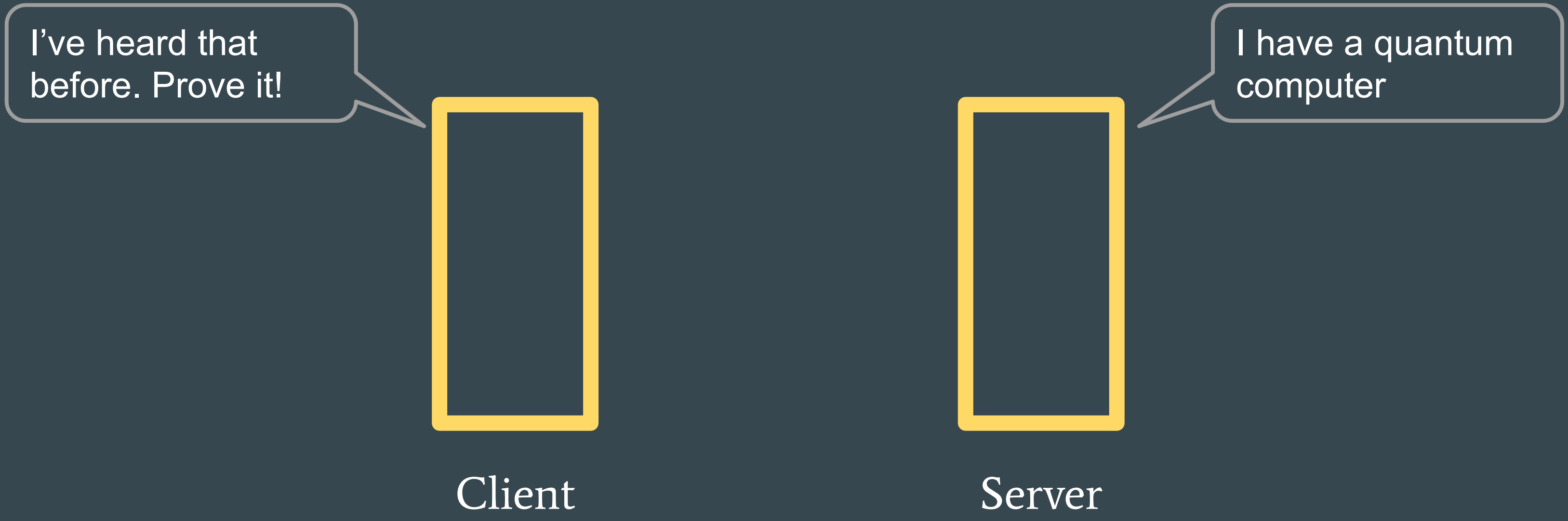
The set of samples which I have in my possession were drawn from a distribution produced by a classical computer in polynomial time

If not, then they must have been implemented by a quantum computer

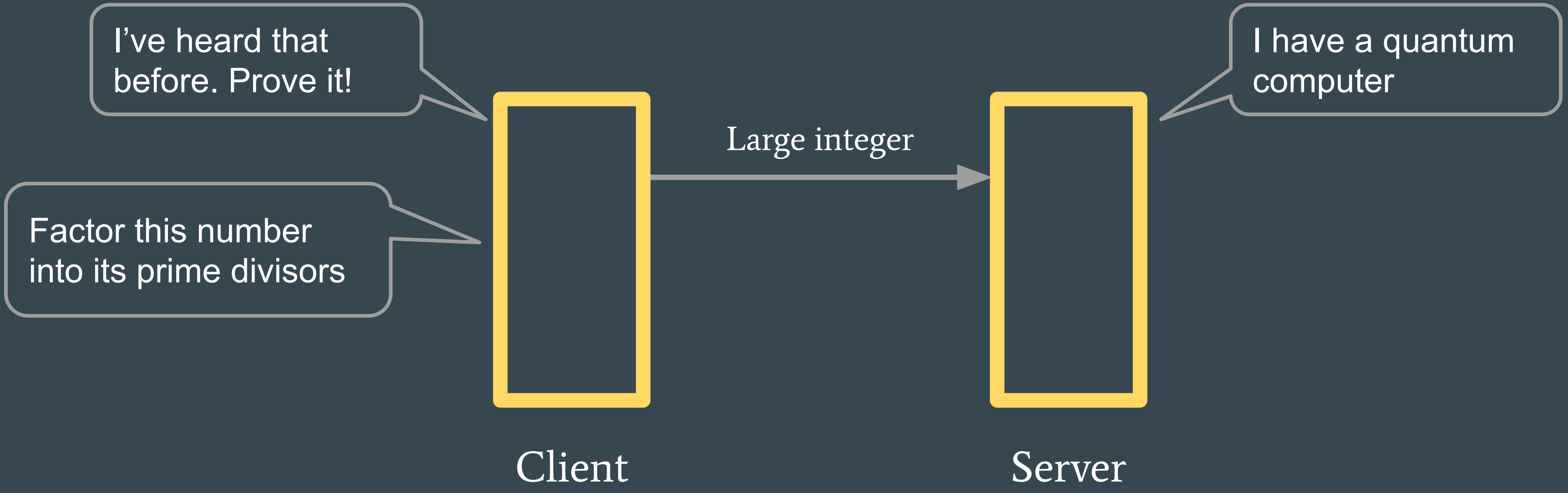
A Proposal



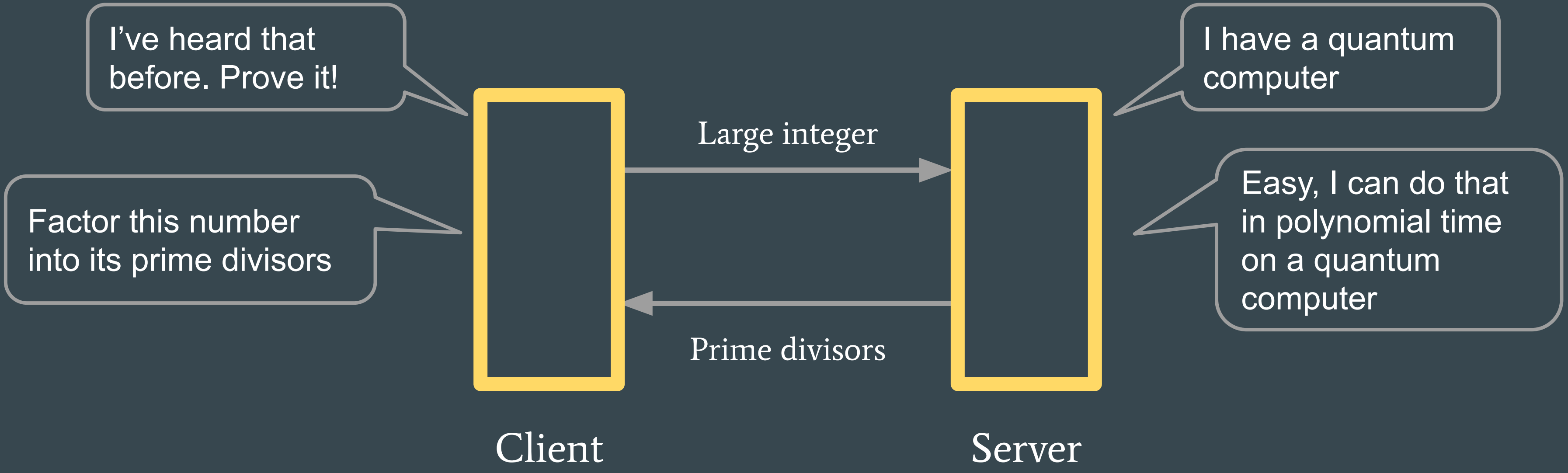
A Proposal


I've heard that before. Prove it!
I have a quantum computer
Client
Server

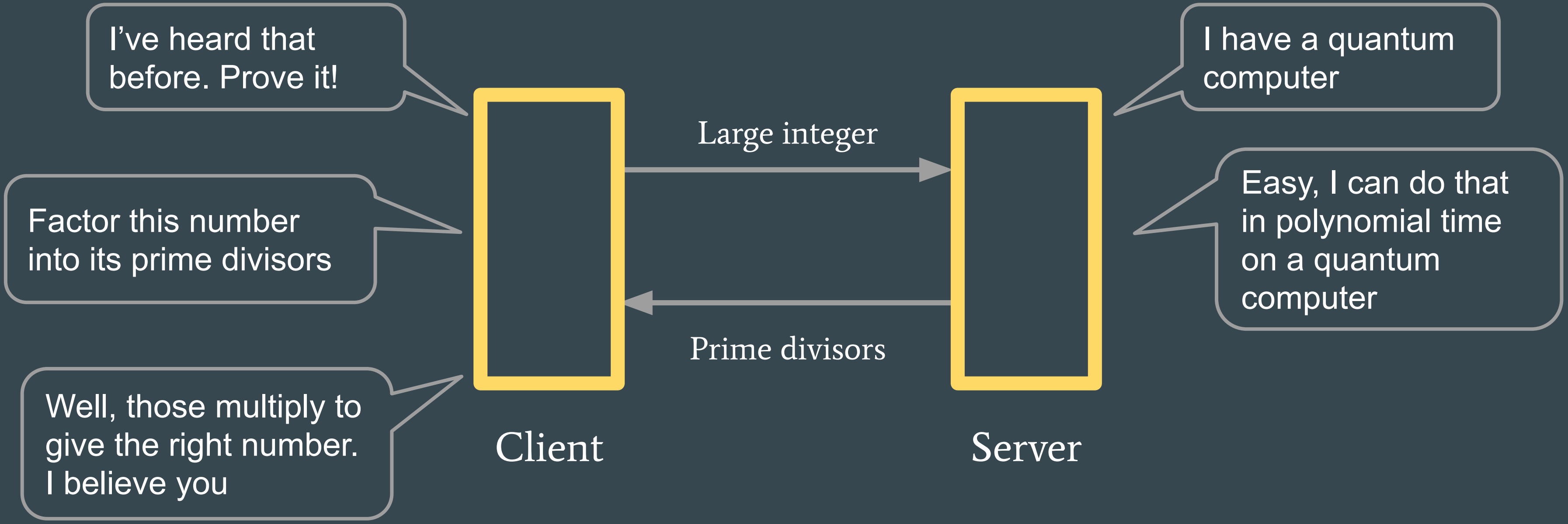
A Proposal


I've heard that before. Prove it!
Factor this number into its prime divisors
Large integer
I have a quantum computer
Client
Server

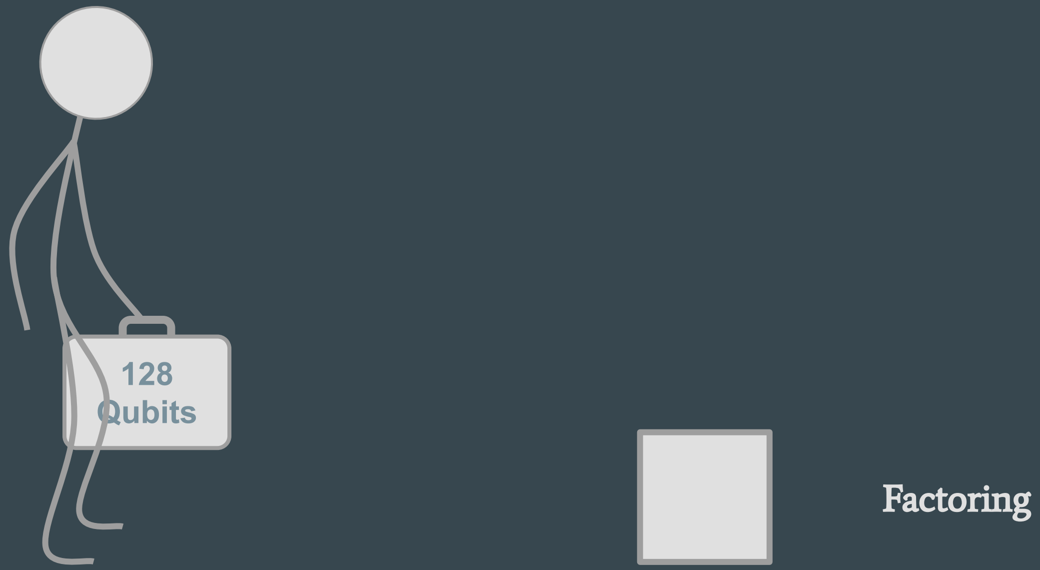
A Proposal


I've heard that before. Prove it!
Factor this number into its prime divisors
Large integer
Prime divisors
Client
Server
I have a quantum computer
Easy, I can do that in polynomial time on a quantum computer

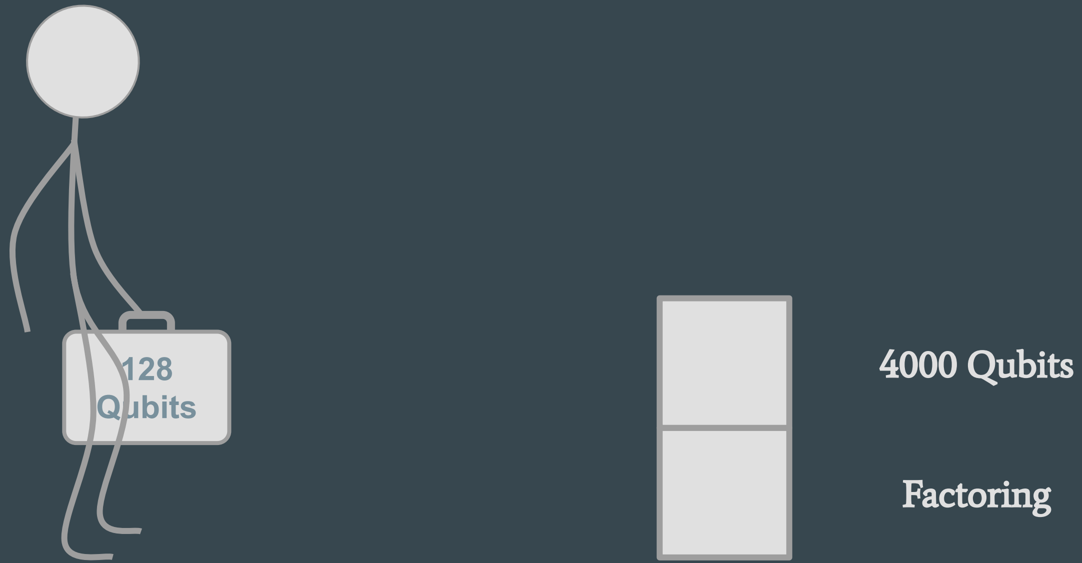
A Proposal


I've heard that before. Prove it!
Factor this number into its prime divisors
Well, those multiply to give the right number. I believe you
Large integer
Prime divisors
Client
Server
I have a quantum computer
Easy, I can do that in polynomial time on a quantum computer

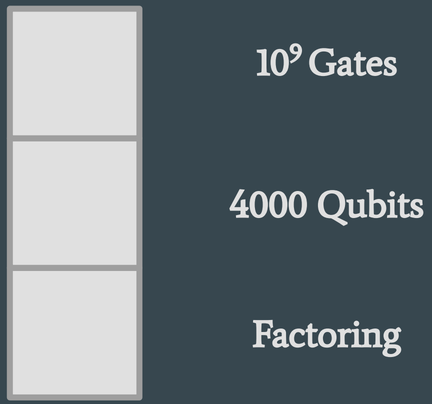
However…



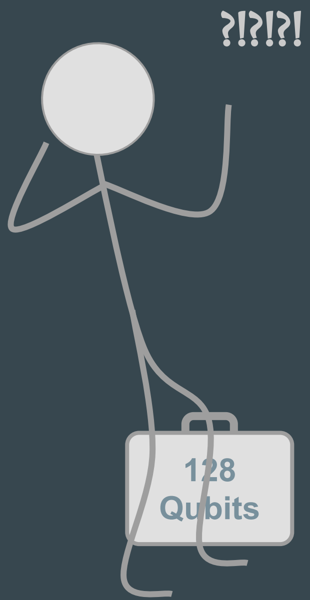
However...

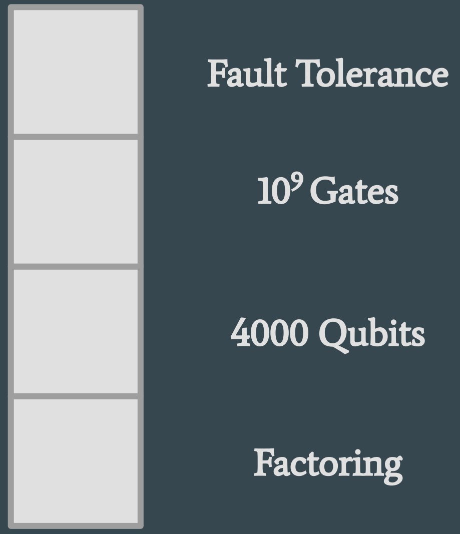


However...

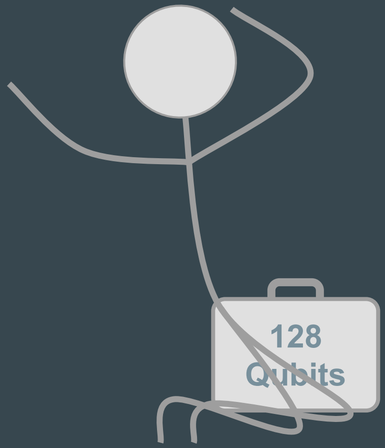


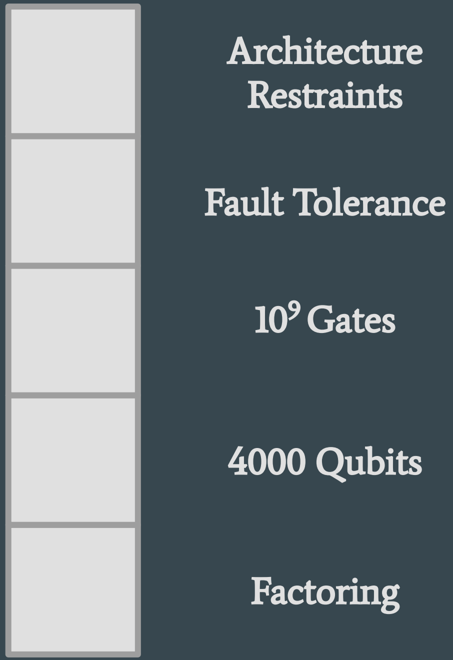
However…


?!?!?!
128
Qubits

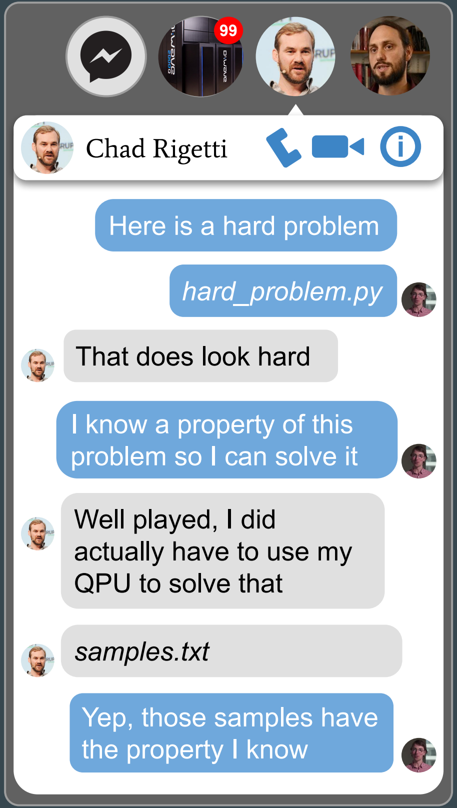

Fault Tolerance
10^9 Gates
4000 Qubits
Factoring

However...


128
Qubits


Architecture Restraints
Fault Tolerance
10^9 Gates
4000 Qubits
Factoring

What Can We Recover From This Failure

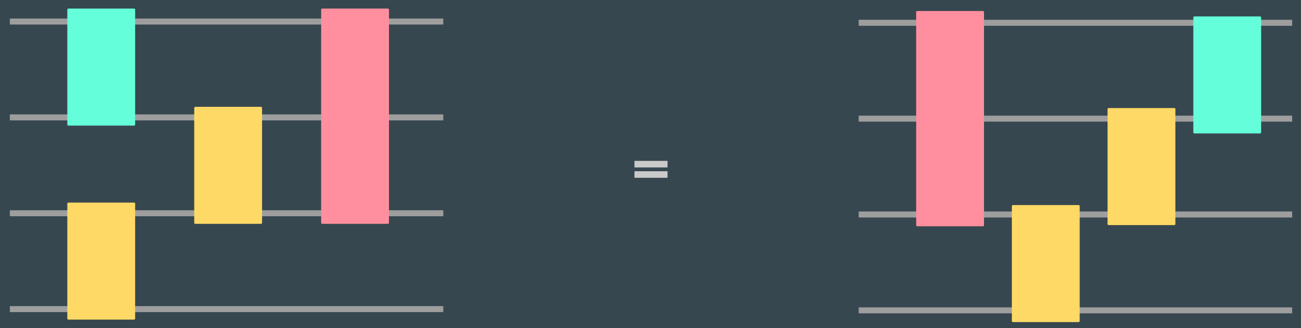


Some Components of the Hypothesis Test to Extract

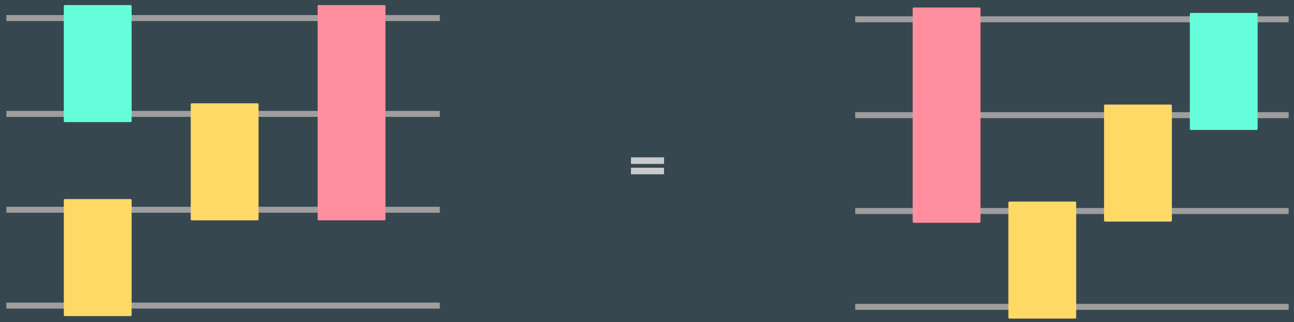
1. A reason Chad must use a quantum computer
   - Hard computational problem
2. Property of the outcome, which is “highly correlated” to the outcome, to check
   - The small hidden problem should be solvable and indicative of the larger problem
3. A backdoor that helps us check property
   - A smaller problem should be hard to uncover
4. Means to implement on NISQ devices
   - Let’s figure something out for IQP... Why not?

IQP as NISQ Device

Circuit Model IQP



Circuit Model IQP



$$\exp\left\{i\theta \bigotimes_{i:q_i=1} X_i\right\}, q \in \{0,1\}^n, \theta \in [0,1]$$

Usually Not True For Quantum Circuits

An Example

$$|+\rangle$$

An Example

“Output qubits”

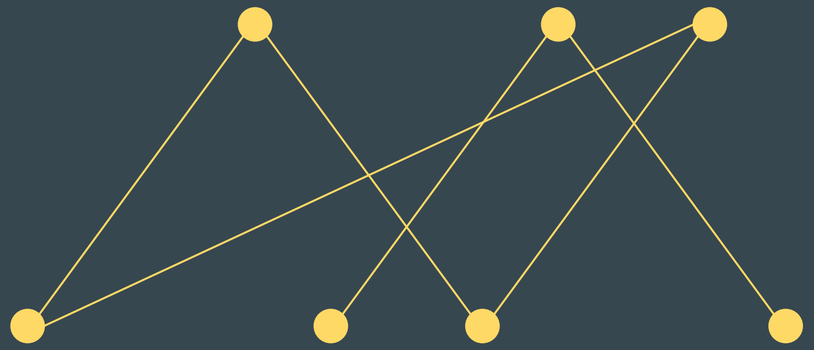
“Gate qubits”

$|+\rangle$

An Example

“Output qubits”

“Gate qubits”

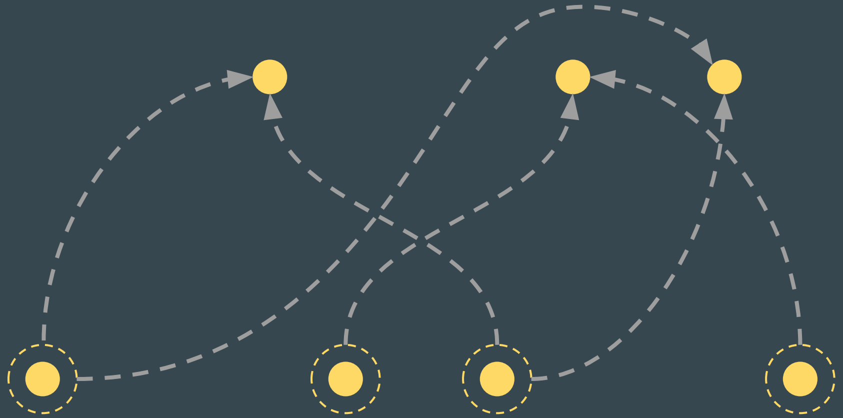


$$cZ \ldots cZ \bigotimes |+\rangle$$

An Example

“Output qubits”

“Gate qubits”

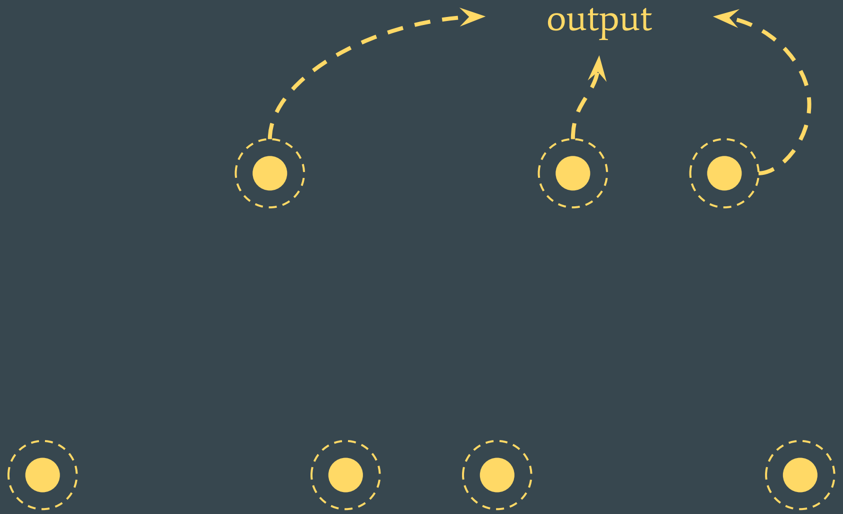


Measurements and classically controlled corrections

An Example

“Output qubits”

“Gate qubits”



Measurements

Advantages And Disadvantages

Advantages:

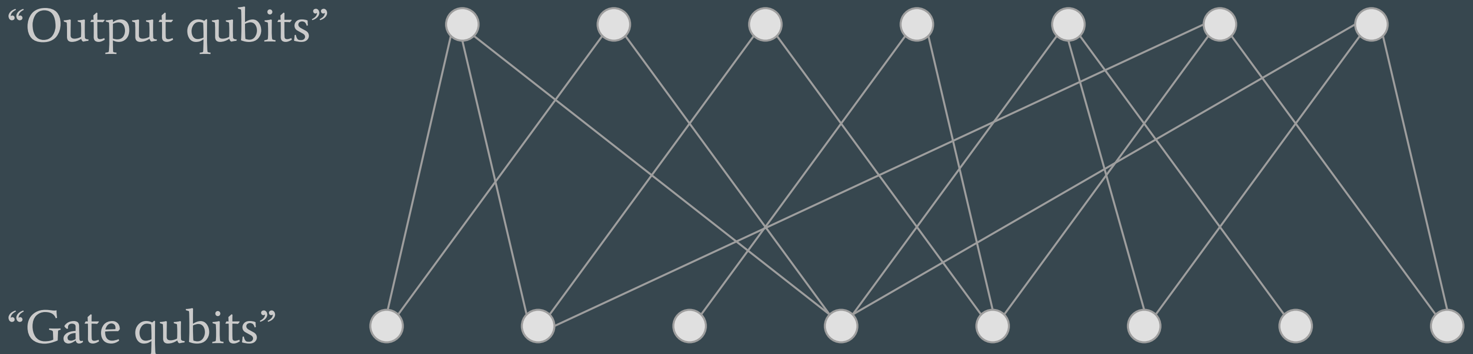
- Can be implemented on NISQ technology
- Believed not be reproducible by a classical computer

Disadvantages:

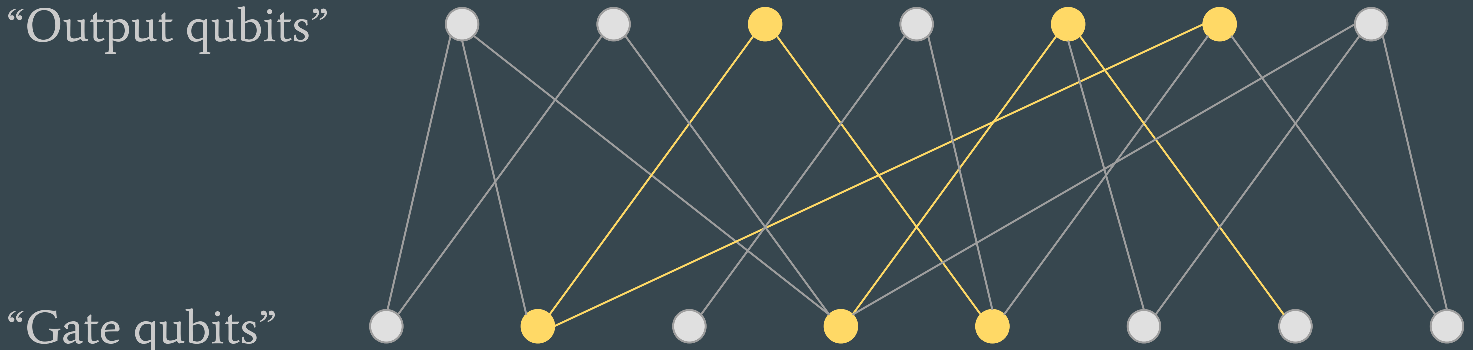
- Not capable of implementing all computations

IQP Hypothesis Test

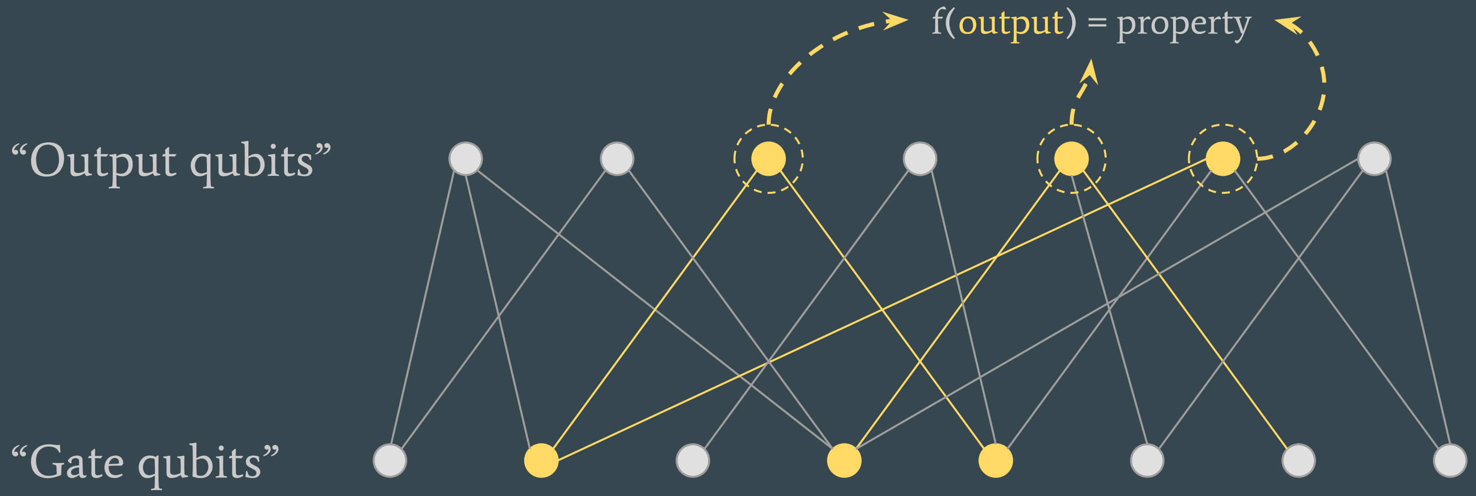
An Example



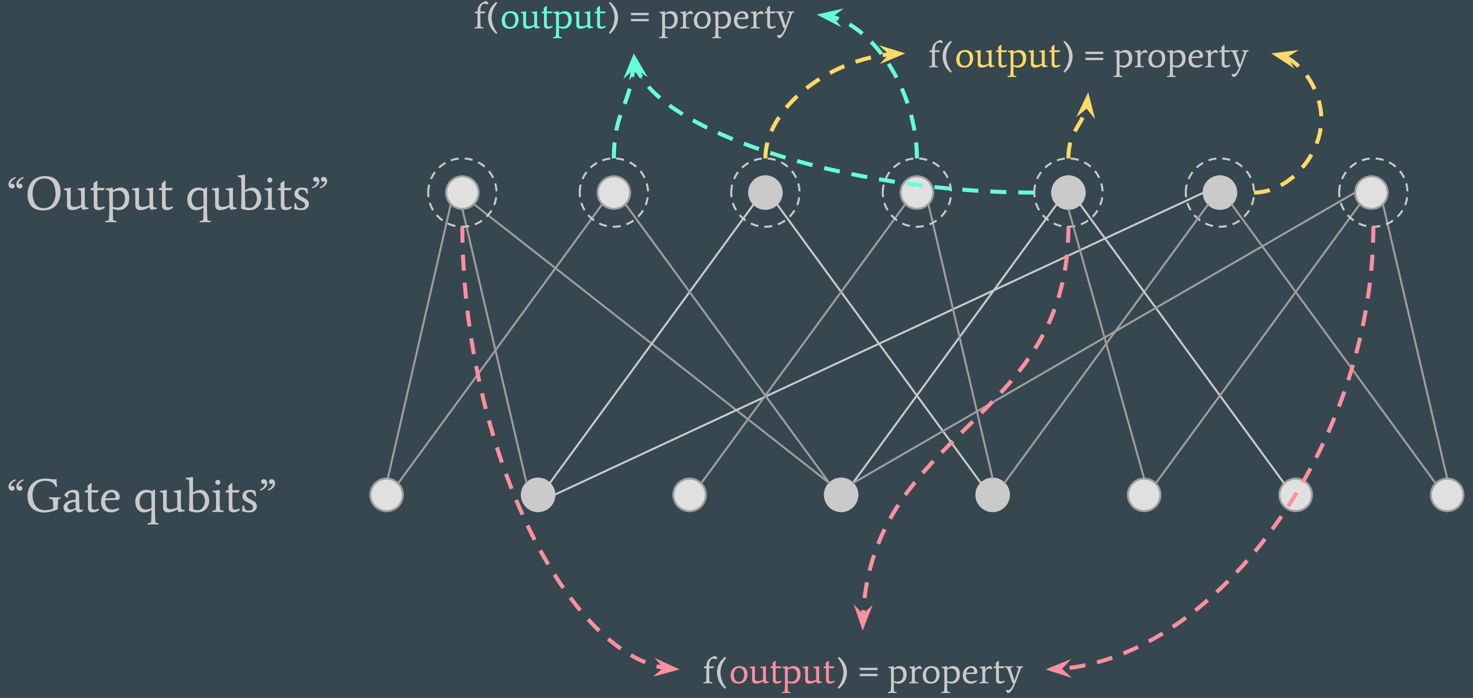
An Example



An Example

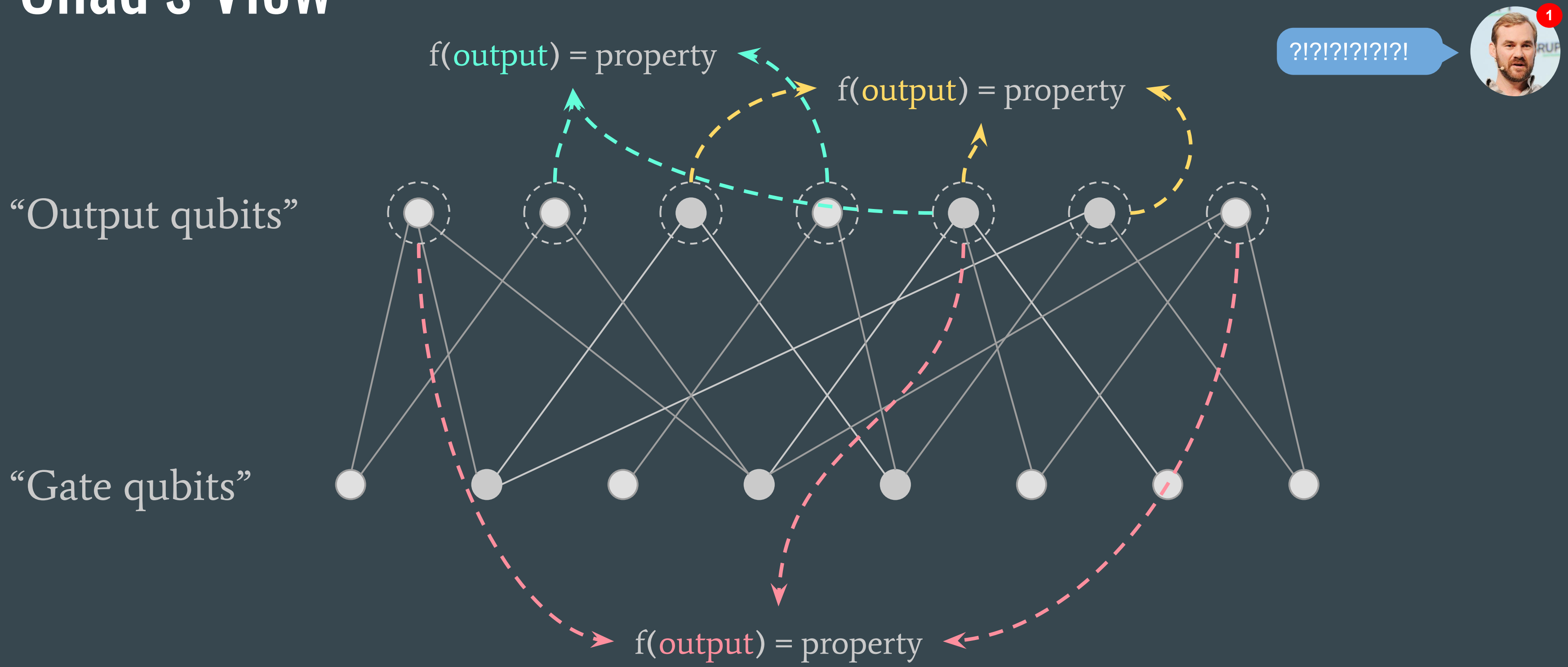


Chad's View



f(output) = property
f(output) = property
"Output qubits"
"Gate qubits"
f(output) = property

Chad's View



f(output) = property
f(output) = property
?!?!?!?!?!?!
1
"Output qubits"
"Gate qubits"
f(output) = property

It Meets The Requirements?

1. A reason Chad must use a quantum computer
    - It looks like a big IQP computation to him
    - Cannot reproduce classically as hiding is good
2. Property of the outcome, which is "highly correlated" to the outcome, to check
    - The property of the hidden graph is fixed so can be checked
    - Its embedding in the larger graph makes it highly correlated
3. A backdoor that helps us check property
    - You know where the small problem is!
4. Means to implement on NISQ devices
    - IQP is easier to implement than BQP

Benchmarking

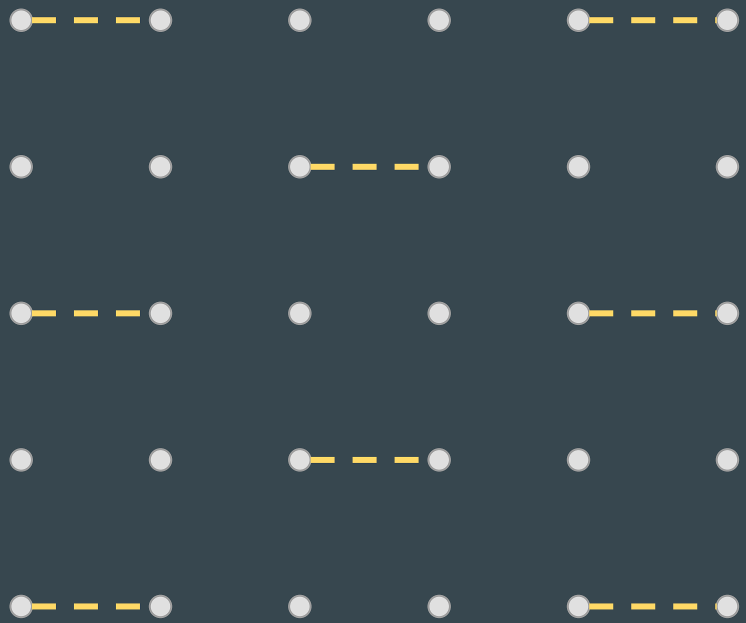
Is The Noise Level Reasonable

Random Circuit

```
Cycle of Hadamard gates
For d clock cycles:
    Apply CZs
    If no CZ applied
        If no random gate acted yet
            Apply T
        Else
            Apply gate different from previous
```

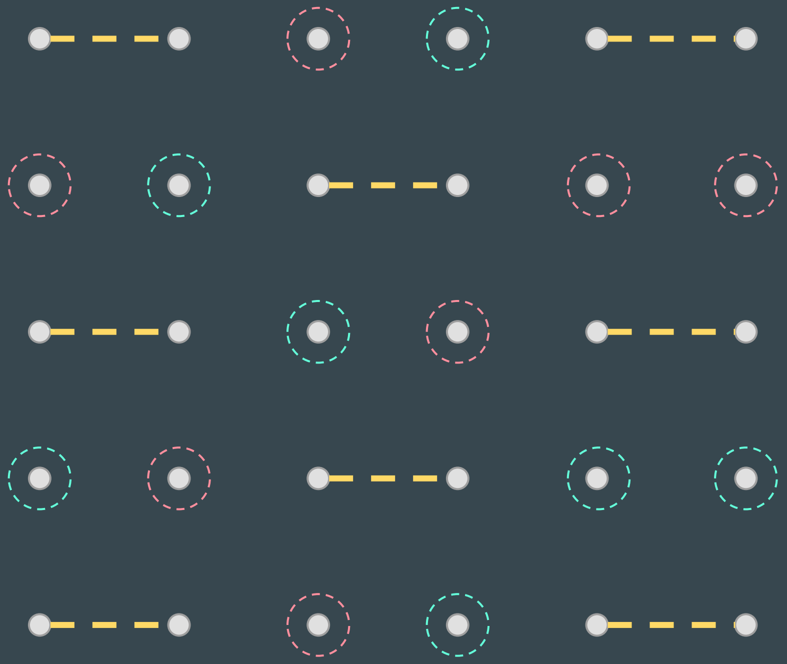
Random Circuit

```
Cycle of Hadamard gates
For d clock cycles:
    Apply CZs
    If no CZ applied
        If no random gate acted yet
            Apply T
        Else
            Apply gate different from previous
```



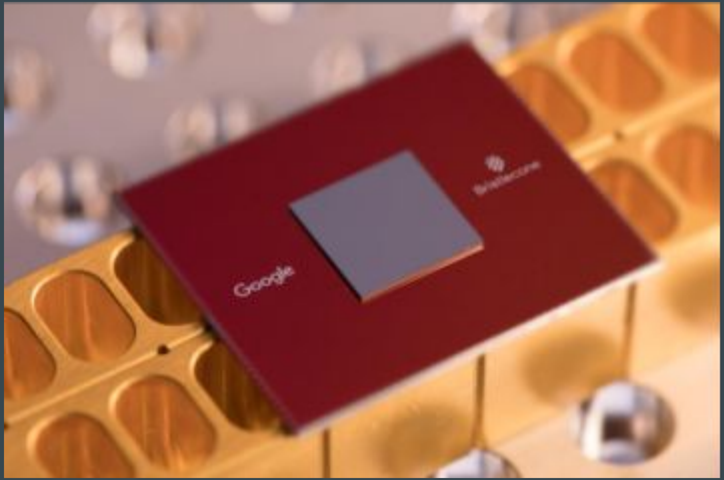
Random Circuit

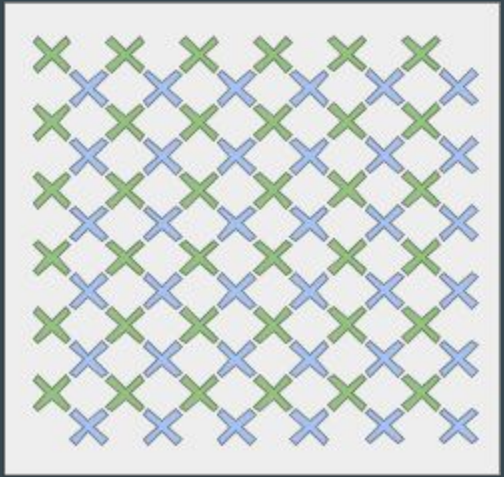
```
Cycle of Hadamard gates
For d clock cycles:
        Apply CZs
        If no CZ applied
                If no random gate acted yet
                        Apply T
                Else
                        Apply gate different from previous
```



Advantages

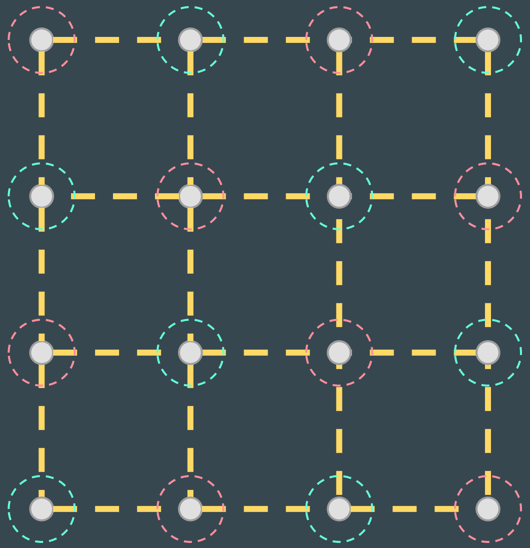
- Designed for Google's Bristlecone device



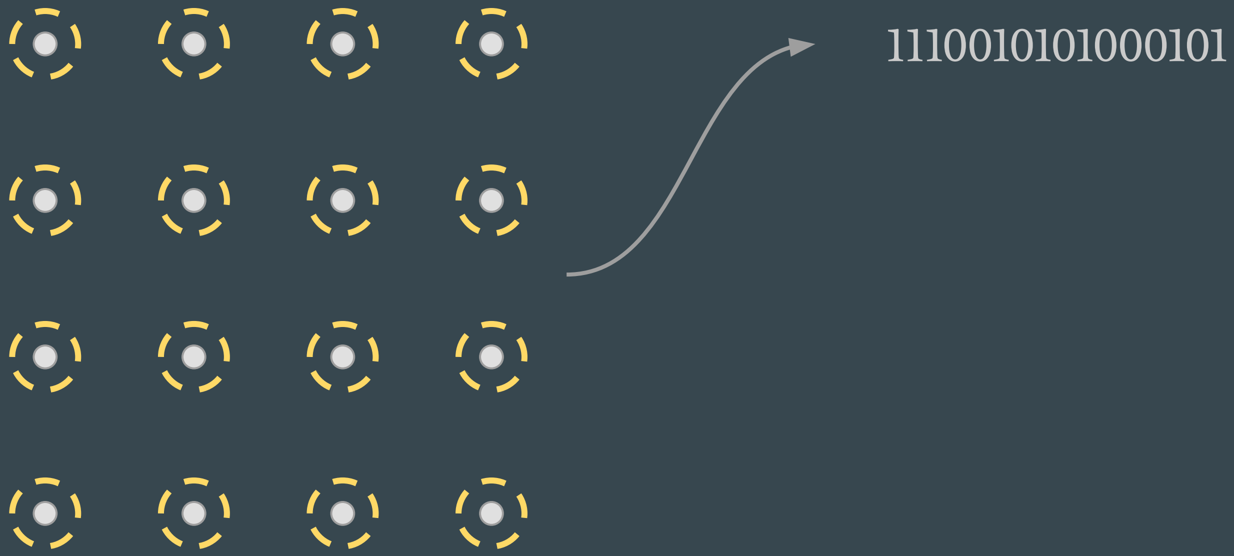


- Also thought to be hard to reproduce on a classical computer

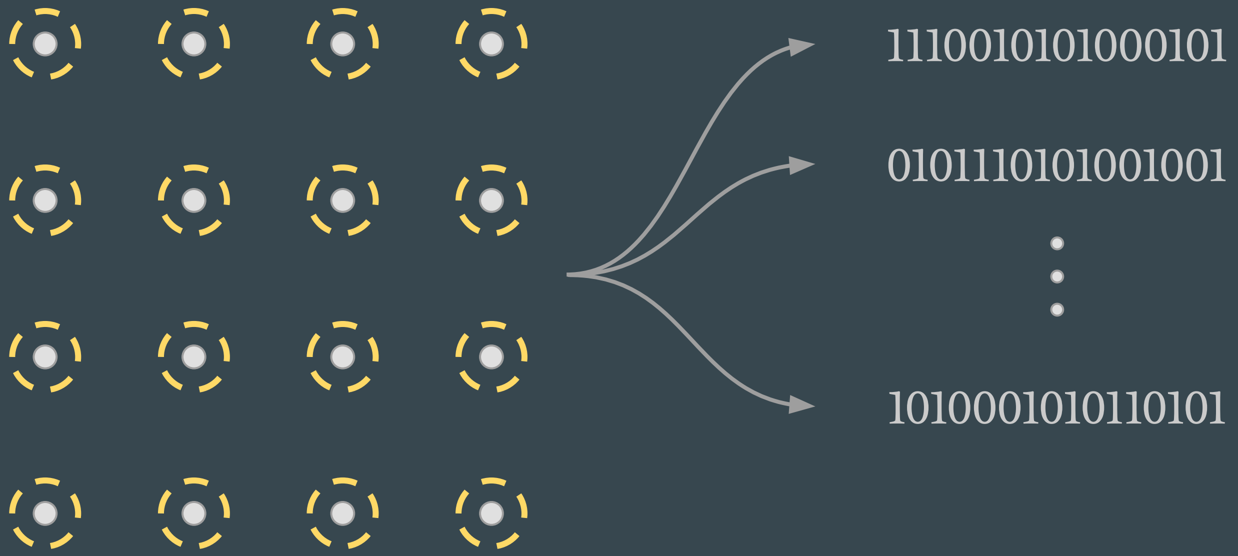
Sampling Problem



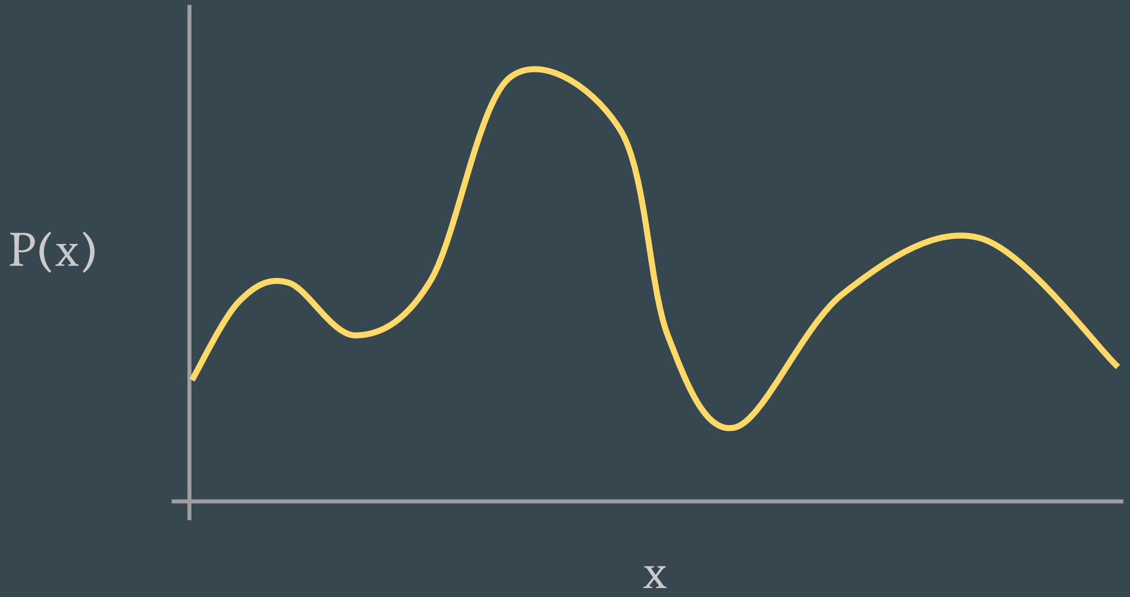
Sampling Problem



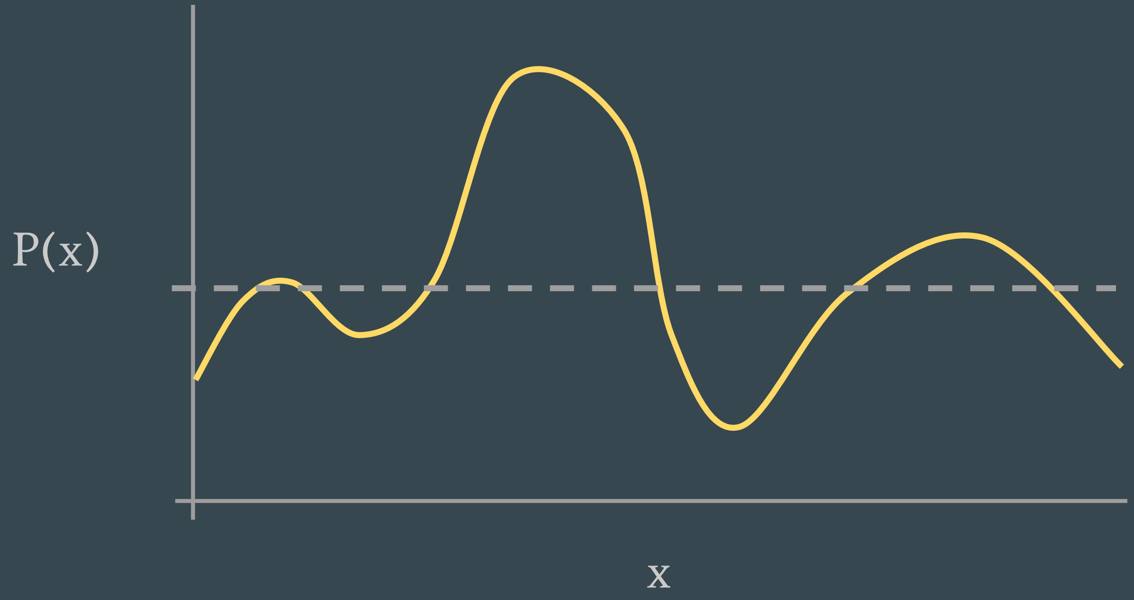
Sampling Problem



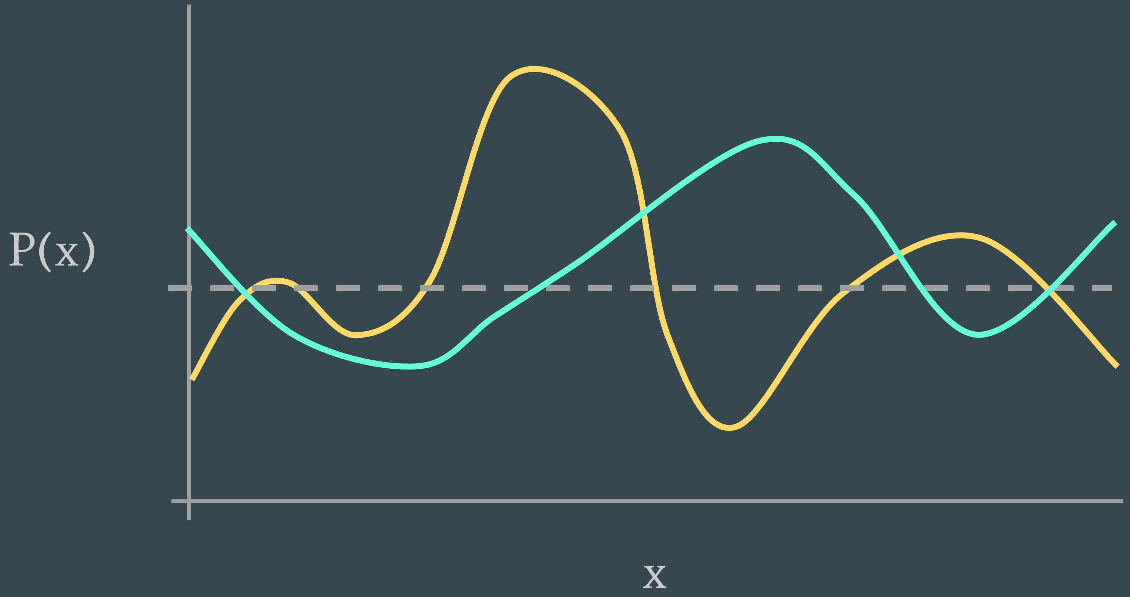
Sampling Output Probability Distribution



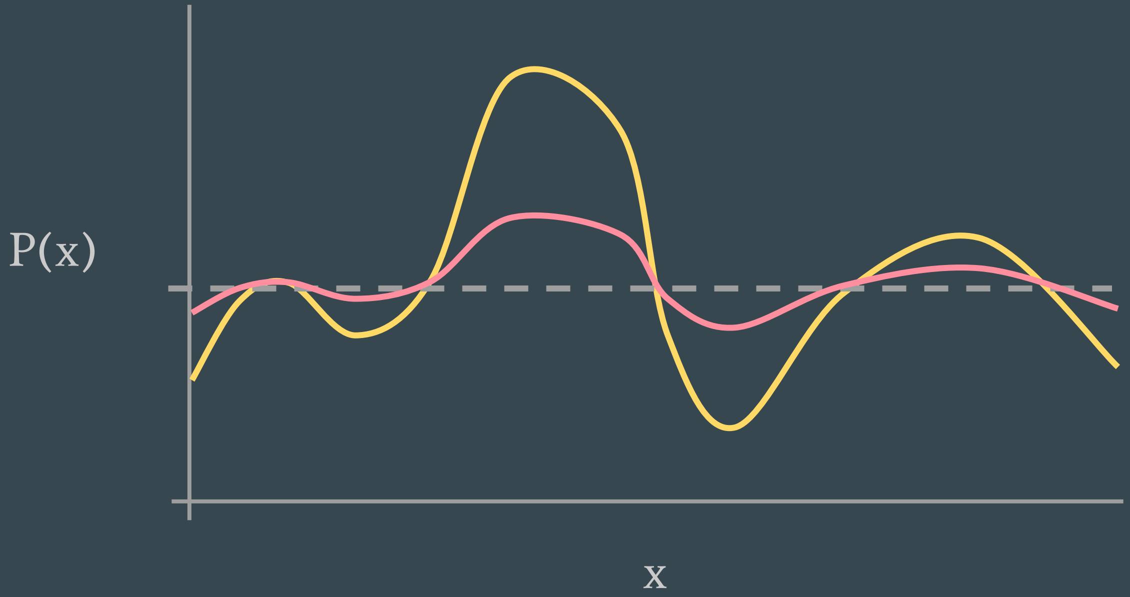
Sampling Output Probability Distribution



Sampling Output Probability Distribution



Sampling Output Probability Distribution



Heavy Output Generation

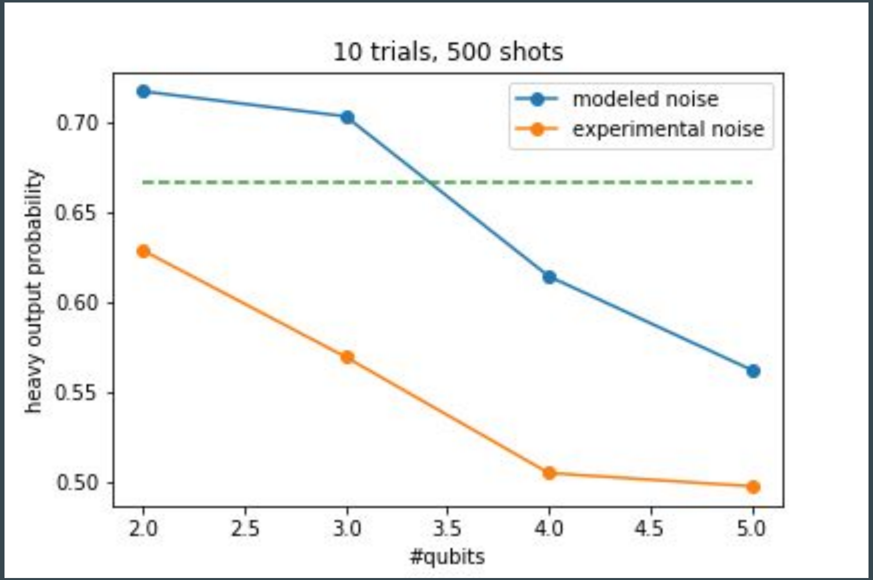
Given as input a random quantum circuit C, generate output strings x_1, ... , x_k at least ⅔ fraction of which have greater than median probability in C's output distribution.

Can be verified in classical exponential time by calculating ideal probabilities

Quantum Volume

- Can a quantum device produce heavy outputs?
- For what size of circuits can the device produce heavy outputs?
- Roughly a measure of number of good qubits

Quantum Volume With Circuit Size



Advantages and Disadvantages

Advantages:

- Global property of device
- Measurable on NISQ devices
- Requires only few sample from the device

Disadvantages:

- Requires exponential resources on a classical computer
- Does not relate to more common complexity results
- While the task is thought to be hard, the grounds for this belief are not as stable

Cross Entropy Difference

Measure quality as the difference from uniform classical sampler

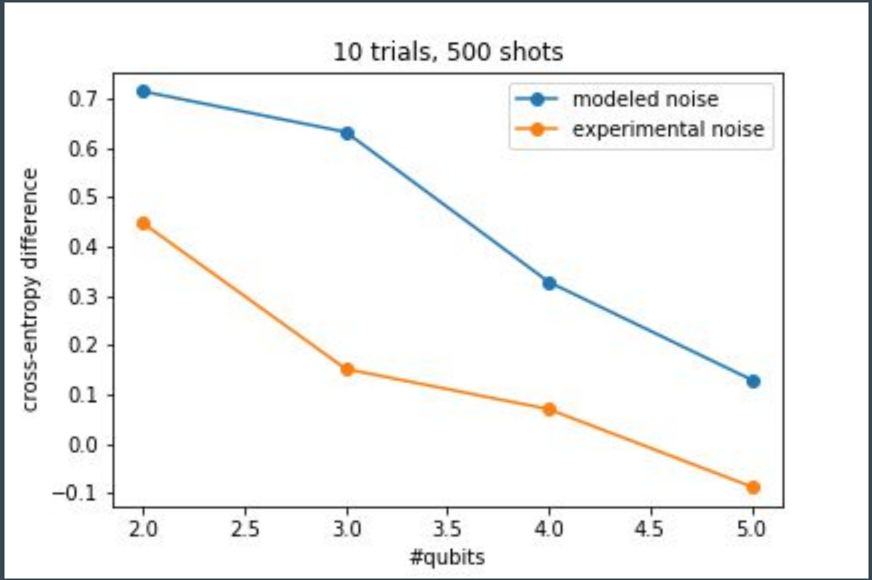
$$\Delta H\left(p_A\right) = \sum_j \left(\frac{1}{N} - p_A\left(x_j|U\right)\right) log \frac{1}{p_U\left(x_j\right)}$$

- Unity for ideal implementation
- Zero for uniform distribution

Achiever supremacy in range:

$$1 \geq \Delta\text{cross-entropy} > C$$

Cross-Entropy With Circuit Size



Conclusions

What Have We Learned

- Hypothesis tests are used to prove “quantumness”
- Benchmarking used to test noise levels

Open Problems

- Does not seem to be a reason to restrict to Random Circuits
  - Or maybe...
  - Random circuits are very flexible
- Can we use these hypothesis tests as a kind of "*meaningful*" verification
- What do hypothesis test teach us about limits of classical computers
  - Where will we see superiority
- Can we benchmark in polynomial time

Thanks!