Verification of NISQ Devices

From Benchmarking to Protocol Verification

Introduction

NISQ

- Few qubits (100-200) - even less
- Limited architecture
- Lots of Noise (I mean really... wow)
  - Verification compensates for lack of error correction
- Verification of sampling
- No fault tolerance and in some cases no error correction

Verification - What do they want?

Physicists

- Certify the outcome of their simulation (ground state/noise)
- Accurately determine physical properties (entanglement/phase estimation/purity)
- Trust in device as “good” quantum simulator in many situations (benchmarks)

Computer scientists

- Verify output of quantum computer is correct (classically intractable)
- Security measures for all situations (best to worst case scenario)
- A bound on trust in your NISQ or UQ device

Industry

- Trust in quantum computer/simulator when involving sensitive/public data
- Assurance that quantum computer/simulator is doing what it should be - efficiency/speed-up?

The public

- “So, if I use a quantum computer to google something it will give me the results even faster and they’ll be better??”
- Are my transactions secure?
- Can we have better drugs and are they safe?

Randomized Benchmarking

What Do You Need And What Can You Get

Requirements

Any amount of qubits (theoretically)

Set of unitaries/gates that form an exact or approximate unitary t-design from which to sample from.

To efficiently run a number of sequence lengths

Inversion of gates or known basis to measure for final state

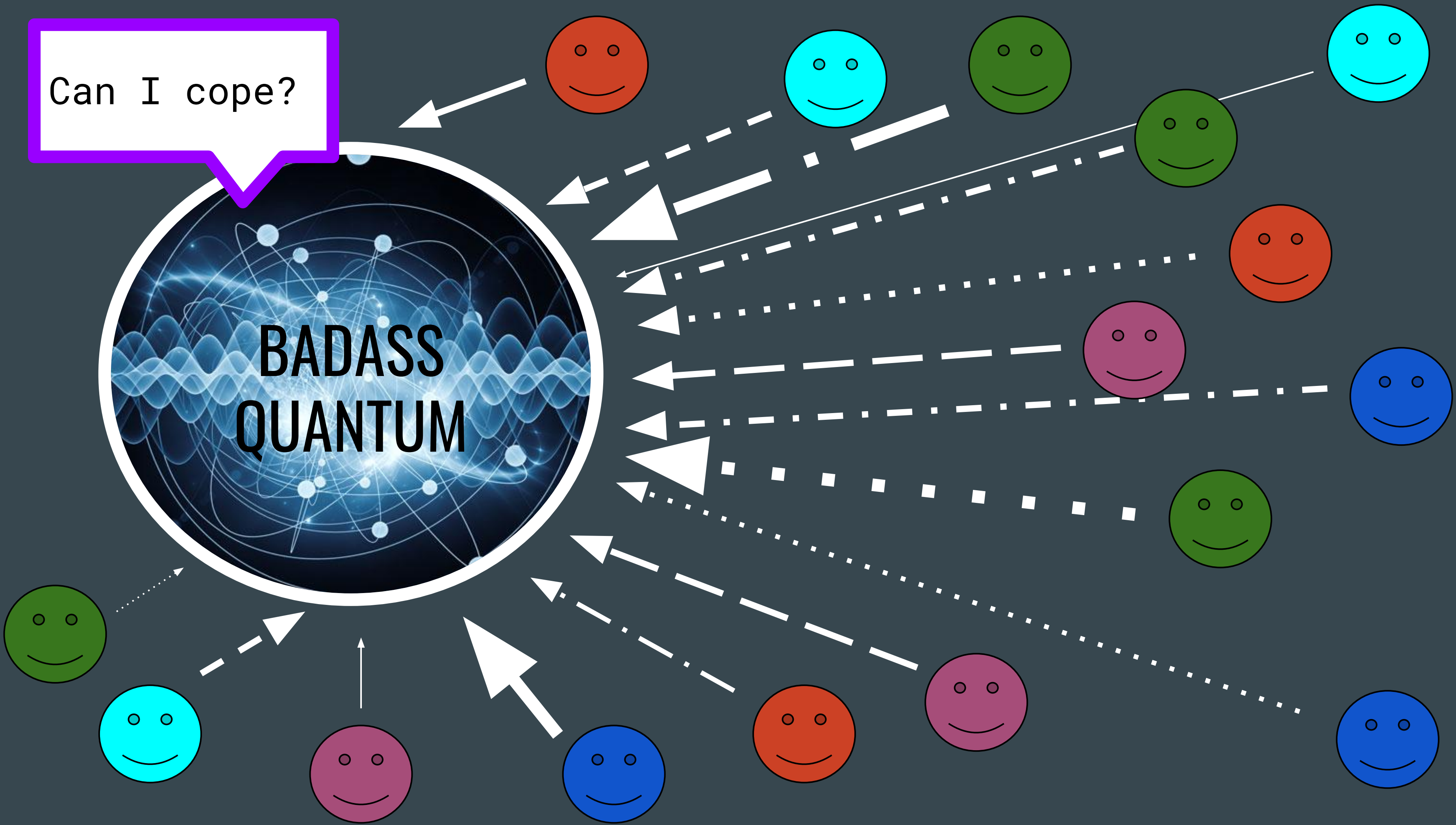
Returns

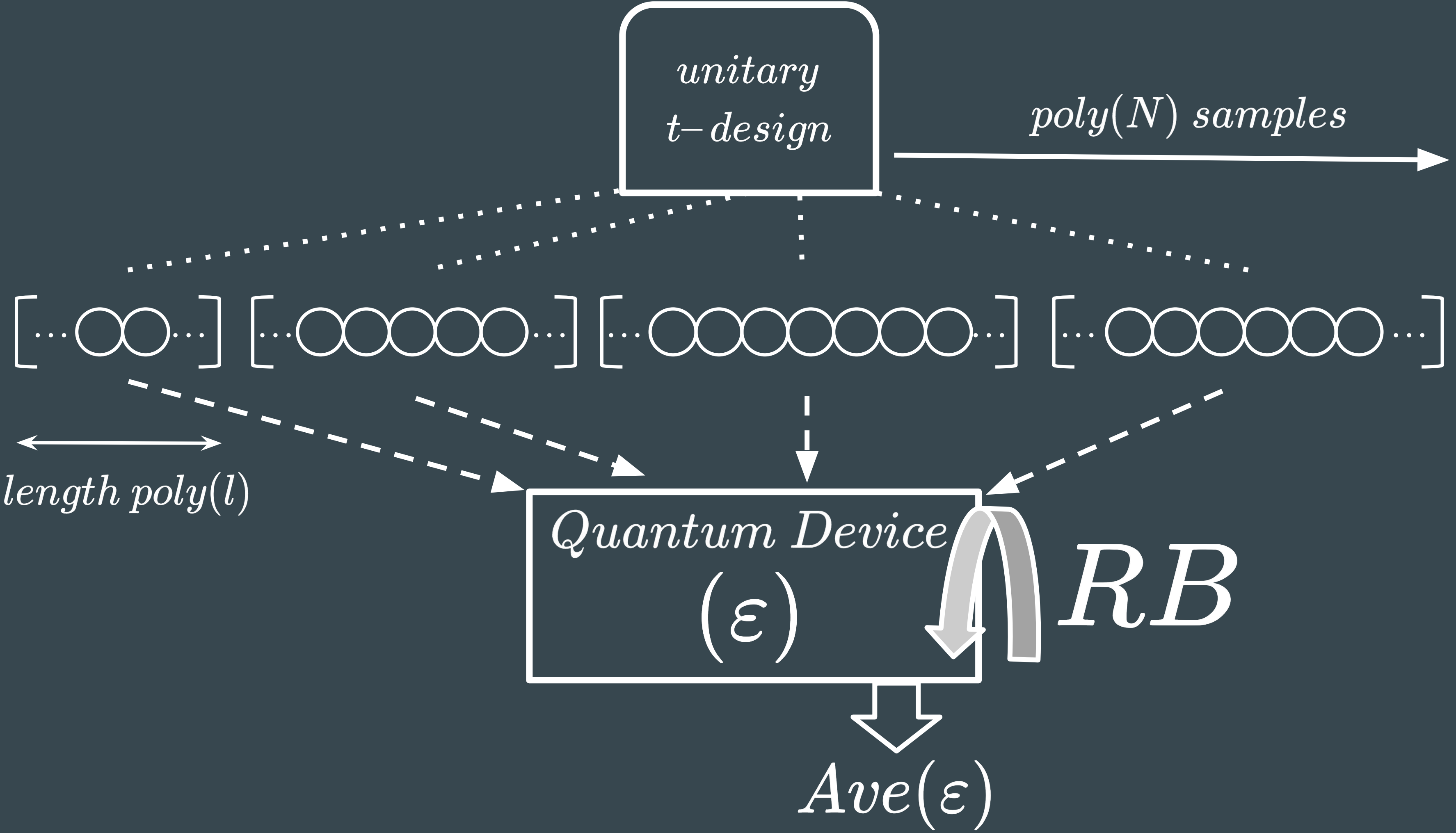
A measure of the average performance of a quantum hardware when running a long quantum information process (partial noise characterisation)

Average error rate of a gateset on your hardware

A measure of a gates performance as a part of a process rather than individually

Incorporates errors from state preparation and measurement


Can I cope?
BADASS
QUANTUM



$unitary$
$t\!-\!design$
$poly(N)\ samples$
$length\ poly(l)$
$Quantum\ Device$
(ε)
RB
$Ave(\varepsilon)$

Fundamentals

Twirling

$$\begin{aligned}\overline{\Lambda}(\rho) &= \int_{U(D)} d\mu(U) U \circ \Lambda \circ U^\dagger(\rho) \\ &= \int_{U(D)} d\mu(U) U \Lambda(U^\dagger \rho U) U^\dagger\end{aligned}$$

Average Λ under the composition $U \circ \Lambda \circ U^\dagger$ for unitary operations $U(\rho) = U\rho U^\dagger$ chosen according to probability distribution $d\mu$

If $d\mu$ is the Haar distribution then the twirled channel on ρ is a depolarising channel

Depolarising Channel

$$\overline{\Lambda}(\rho) = p\rho + (1-p)\frac{1}{D}$$

Strength of channel

Fundamentals

Unitary t-design

Twirling

$$\overline{\Lambda}(\rho) = \int_{U(D)} d\mu(U) U \circ \Lambda \circ U^\dagger(\rho)$$
$$= \int_{U(D)} d\mu(U) U \Lambda(U^\dagger \rho U) U^\dagger$$

Average Λ under the composition $U \circ \Lambda \circ U^\dagger$ for unitary operations $U(\rho) = U \rho U^\dagger$ chosen according to probability distribution $d\mu$

If $d\mu$ is the Haar distribution then the twirled channel on ρ is a depolarising channel

Depolarising Channel

$$\overline{\Lambda}(\rho) = p\rho + (1-p)\frac{1}{D}$$

Strength of channel

Fidelity

$$F(\Lambda_U, U) = F(U|\psi\rangle\langle\psi|U^\dagger, \Lambda_U(|\psi\rangle\langle\psi|))$$

$$F(\Lambda_U, U) = F(\Lambda_{U,e}, I) \longrightarrow F(\textstyle\int_{Haar} \Lambda_{U,e}, I)$$

$$\Lambda_U(X) = \textstyle\sum_i A_i X A_i^\dagger$$

$$\Lambda_U(X) = \textstyle\sum_i (A_i U^\dagger) U X U^\dagger (U A_i^\dagger)$$

$$\Lambda_U = \Lambda_{U,e} \circ U$$

$$\Lambda_{U,e} = \textstyle\sum_i A_i U^\dagger \otimes U A_i^\dagger$$

Fidelity

$$F(\Lambda_U, U) = F(U|\psi\rangle\langle\psi|U^\dagger, \Lambda_U(|\psi\rangle\langle\psi|))$$

$$F(\Lambda_U, U) = F(\Lambda_{U,e}, I) \xrightarrow{\hspace{3cm}} F(\textstyle\int_{Haar} \Lambda_{U,e}, I)$$

$$\textstyle\int_U d\mu(U) F(\Lambda_U, U) = \int_U d\mu(U) F(\Lambda_{U,e}, I)$$

$$\Big\downarrow$$

$$(1 - \tfrac{D-1}{D}) + \tfrac{D-1}{D}(1 - \overline{p_d})$$

$$\Lambda_U = \Lambda_{U,e} \circ U \xrightarrow{\hspace{3cm}} \Lambda_{U,e} = \Lambda_U \circ U^{-1}$$

Method

$$\rho = |\psi\rangle\langle\psi|$$

REPEAT

$$M = \{E, 1 - E\}$$

Average survival probability

$$\Lambda^{-1}_{U_{Tot}} \Lambda_{U_m} \dots \Lambda_{U_1} \longrightarrow Tr(ES_m(\rho))$$

- Average each survival probability over number of sequences sampled at that length: average survival probability over all possible sequences at that length
- Do this for varying sequences - where all unitaries are sampled from a unitary t-design.

$$P_m = A + (B + Cm)p^m$$

$$r = \frac{D-1}{D}(1-p)$$

Verified/Secure?

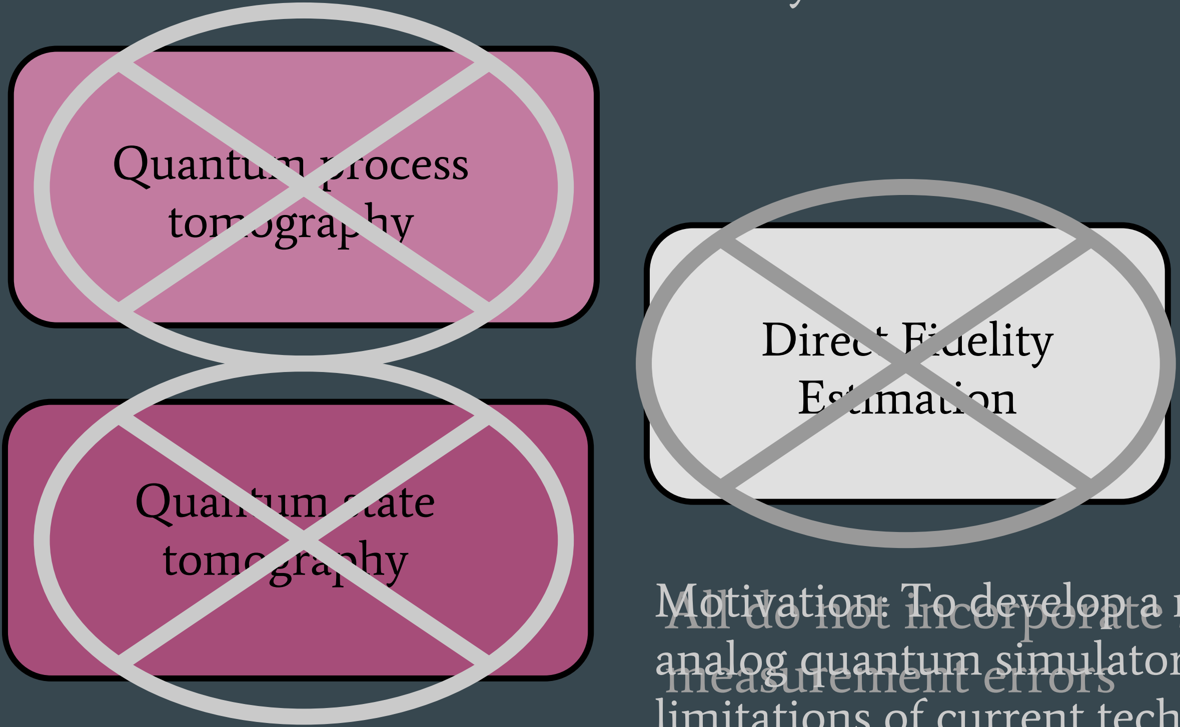
Random processes not specific algorithms - correct outcome of computation **not verified** with this technique

The “server” and “verifier” know the initial state of the system, the random processes run on the device and the measured output - **not secure**

If your specific algorithm were hidden in the random processes somehow, could we get a measure of the average error rate for that process on the hardware without the “server” knowing what the algorithm was?

In the analog setting - why is this interesting ?

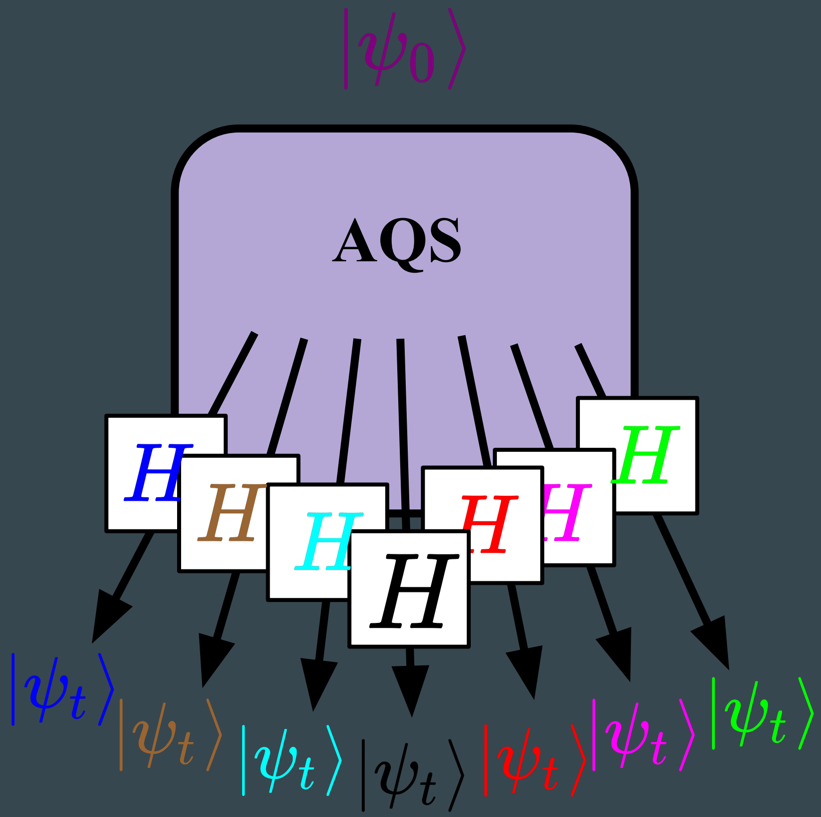
Not efficiently scalable



All do not incorporate state preparation and measurement errors

Motivation: To develop a method for testing analog quantum simulators that goes beyond the limitations of current techniques

Programmable analog quantum simulators



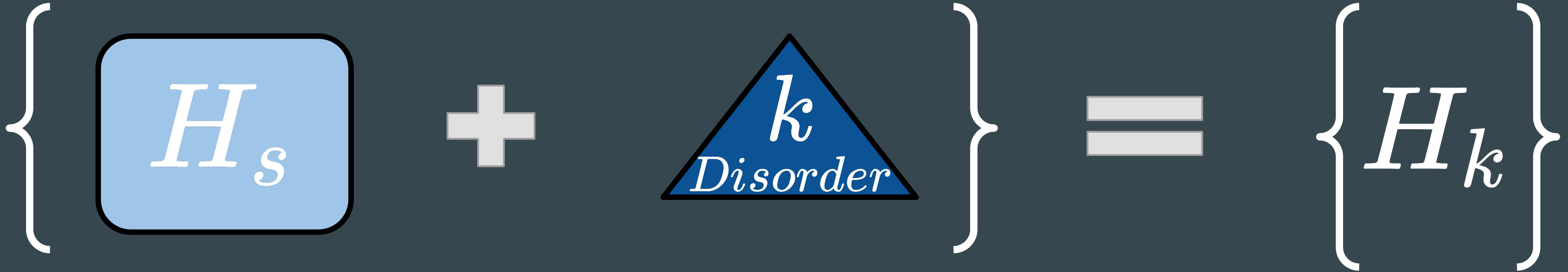
Tunable

Reproducible

For:
Problems that require being able to run a whole class of hamiltonians in a reproducible way.

Way to test/certify such a simulator?

In the analog setting - RB method



$$\left\{ \boxed{H_s} + \underset{\text{Disorder}}{k} \right\} = \left\{ H_k \right\}$$

Set to sample unitaries

$$\left\{ U_k = e^{-iH_k t} \right\}$$

Imperfectly implemented: $\Lambda_{U_k} = \varepsilon \circ U_k$

In the analog setting - RB method

Same as standard RB but :

- Unitaries are time-evolution operators sampled from set generated from native gates of system

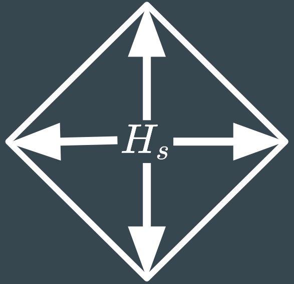
- Each unitary is systematically inverted (for now) rather than one single deterministic inversion operator

Generating a unitary t-design from Hamiltonian

Non trivial problem

Generate disorder around static Hamiltonian - break symmetry enough to generate a unitary t-design - (disorder potential + interaction term)

- Product of those generated will eventually span unitary space



- For 2-design can compare second moment of Haar measure with second moment of unitaries generated : basically compare eigenvalues - should be two max with 1 and 0’s everywhere else

Verification with randomized benchmarking?

- Can we get an average error rate for a specific quantum algorithm?

- Need it to appear random, or be hidden within a random sequence

$$\left\{\begin{matrix} U_{alg} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & U_t \\ U_t & \cdot & \cdot & \cdot \end{matrix}\right\}$$

Embed specific algorithm sequence within sequence of random unitaries from unitary t-design.

- Build unitary t-design around specific algorithm?

Quantum Benchmark (company)

Claims

"The True-Q(™) Validation software system *accurately* validates the Quantum Capacity of *any* quantum hardware platform to execute any quantum circuit for *any* user-supplied problem or application."

"Validates the capacity of *any* quantum hardware platform to perform *any* user-supplied algorithm to *any* user-specified precision"

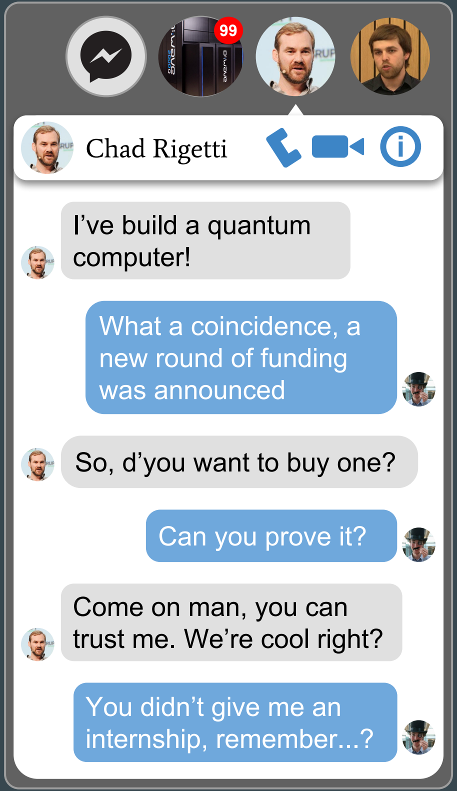
"True-Q(™) Design is a scalable solution for optimizing hardware design and quantum computing performance."

How do they achieve this?

- ***Randomized Benchmarking:*** accurate and precise error characterization of elementary quantum gates
- ***Cycle Benchmarking:*** scalable error characterization of arbitrary parallelized gate cycle and universal (polynomial-depth) quantum circuits
- ***Scalable Error Reconstruction:*** detailed error reconstruction across the quantum processor to find error correlations and optimize hardware design and performance of quantum error correcting codes
- ***Randomized Compiling:*** efficient run-time error suppression for arbitrary applications
- ***Quantum Capacity:*** high-precision performance validation for arbitrary applications

Hypothesis Testing

The Setting



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...?

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

Unlike traditional experiments this amounts to the *nonexistence* of something. Hence we need some theoretical tools to guide us

Boson Sampling

Constant-Depth Quantum Circuits

extended clifford circuits

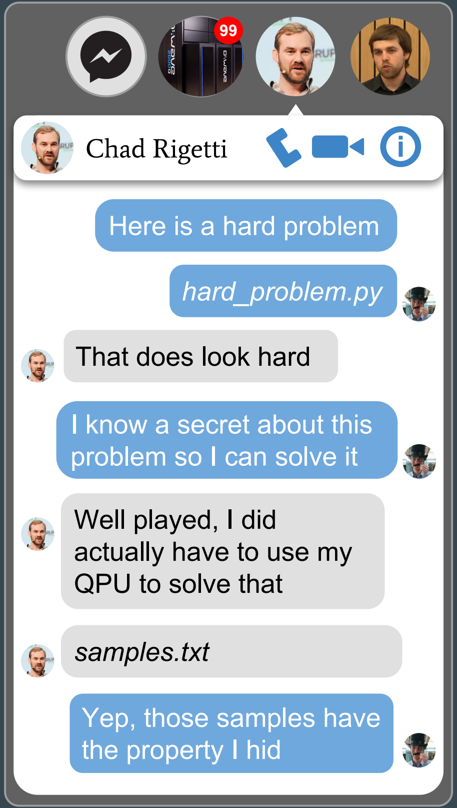
IQP

one-clean-qubit

QAOA

Ball Permutations

One Possible Option



Chad Rigetti

Here is a hard problem

hard_problem.py

That does look hard

I know a secret about this problem so I can solve it

Well played, I did actually have to use my QPU to solve that

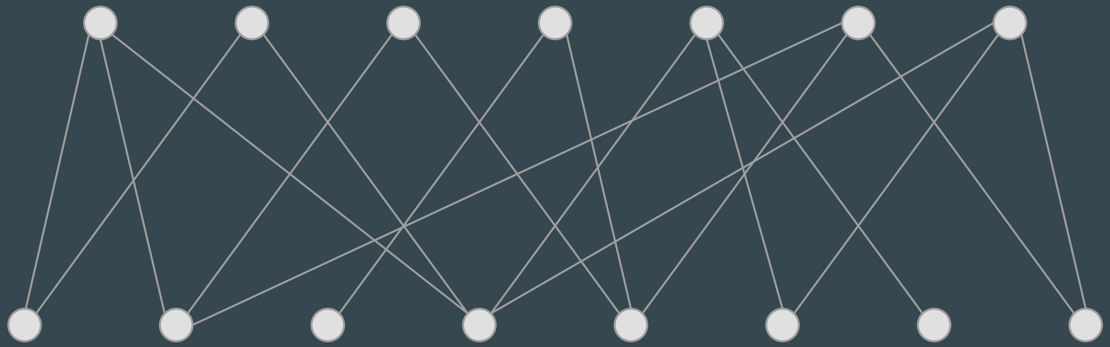
samples.txt

Yep, those samples have the property I hid

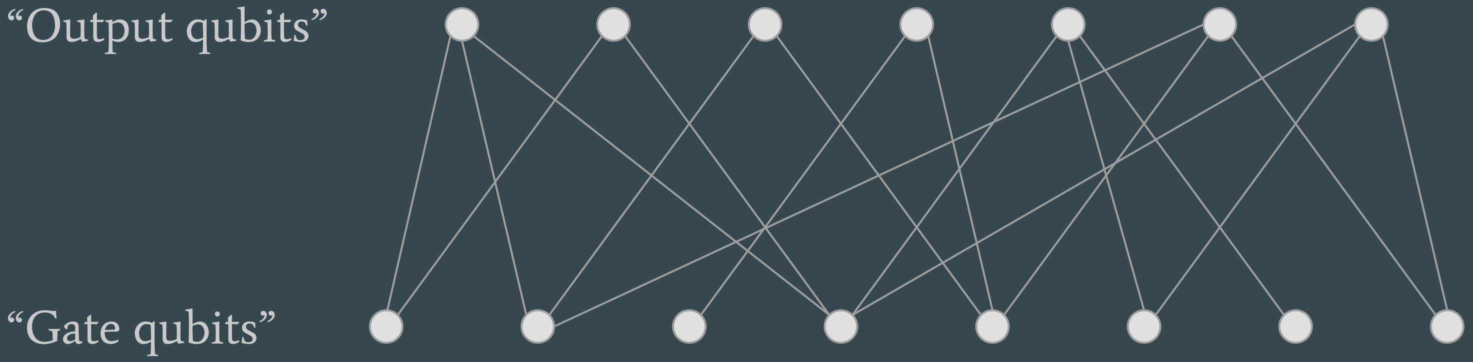
Some Components of the Hypothesis Test to Extract

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?

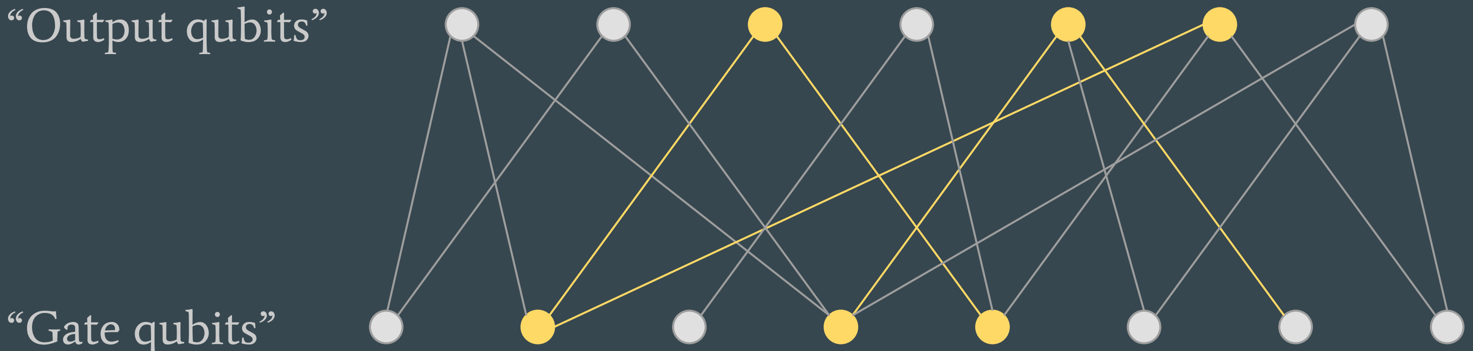
An Example



An Example



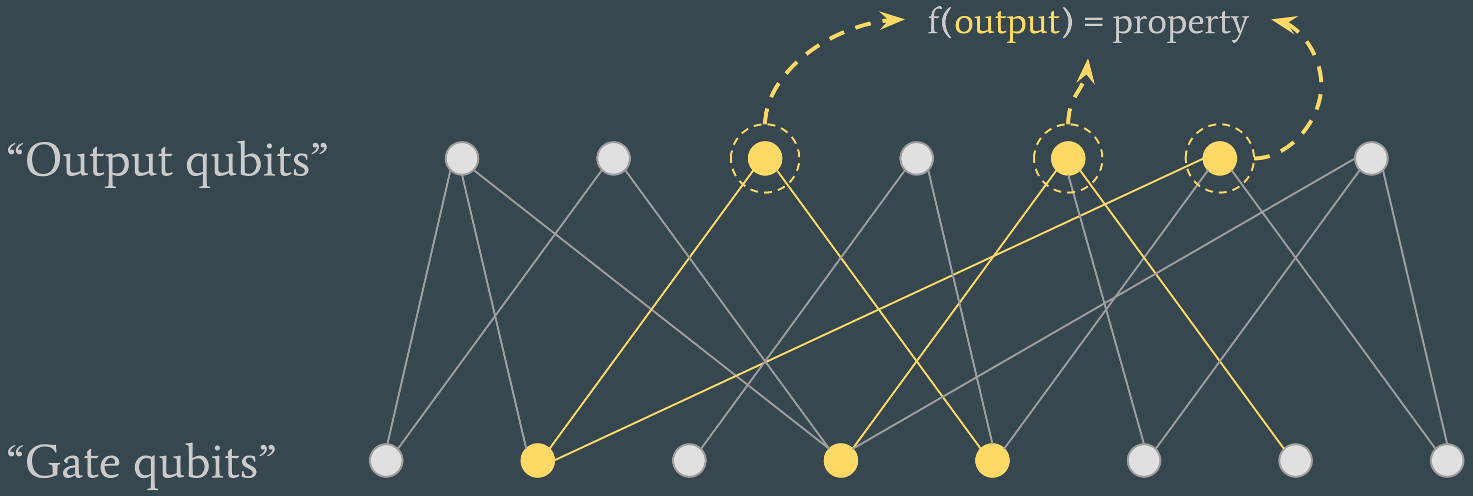
An Example



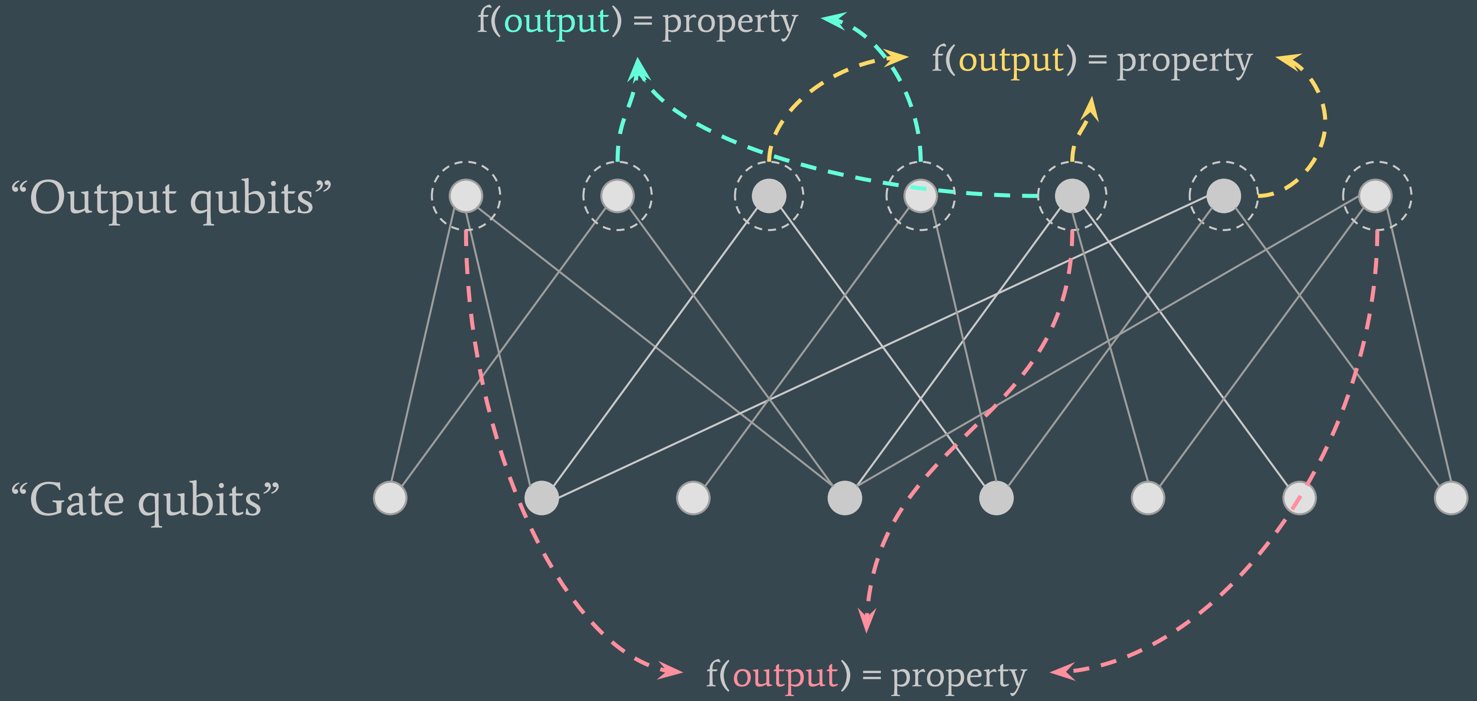
“Output qubits”

“Gate qubits”

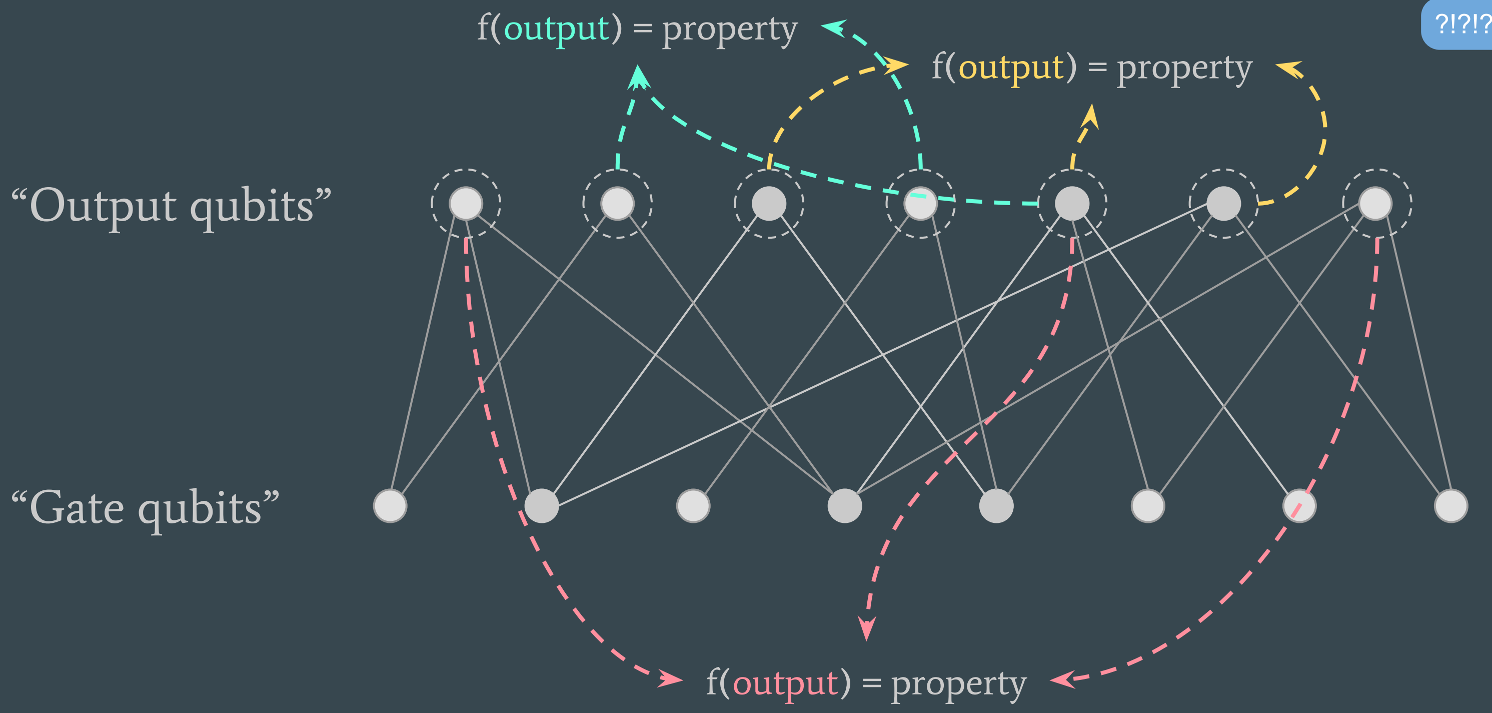
An Example



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

?!?!?!?!?!?!
1

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

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!

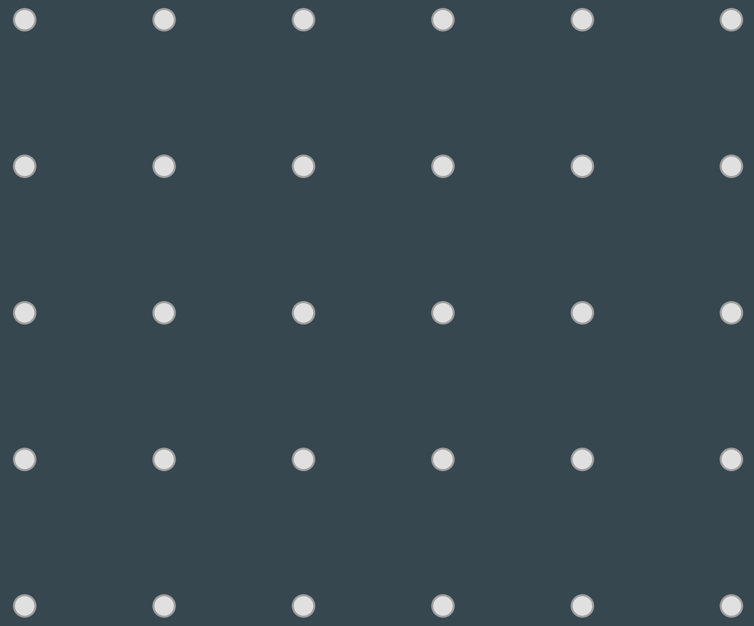
FORBIDDEN

4. Means to implement on NISQ devices
   - ○ IQP is easier to implement than BQP

Random Circuit

For example:

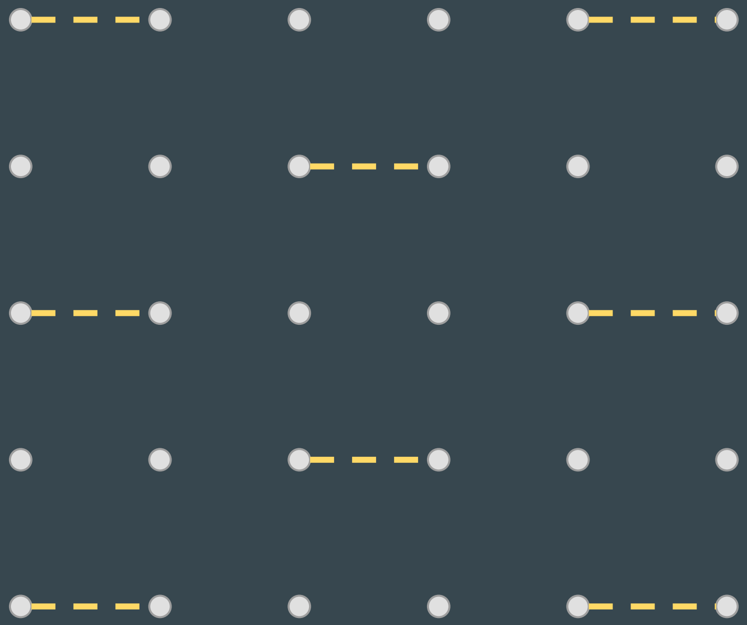
```
Cycle of Hadamard gates
For d clock cycles:
    Apply CZs
    If no CZ applied
        If no random gate acted yet
            Apply T
        Else
            Apply gate different from previous
```



Random Circuit

For example:

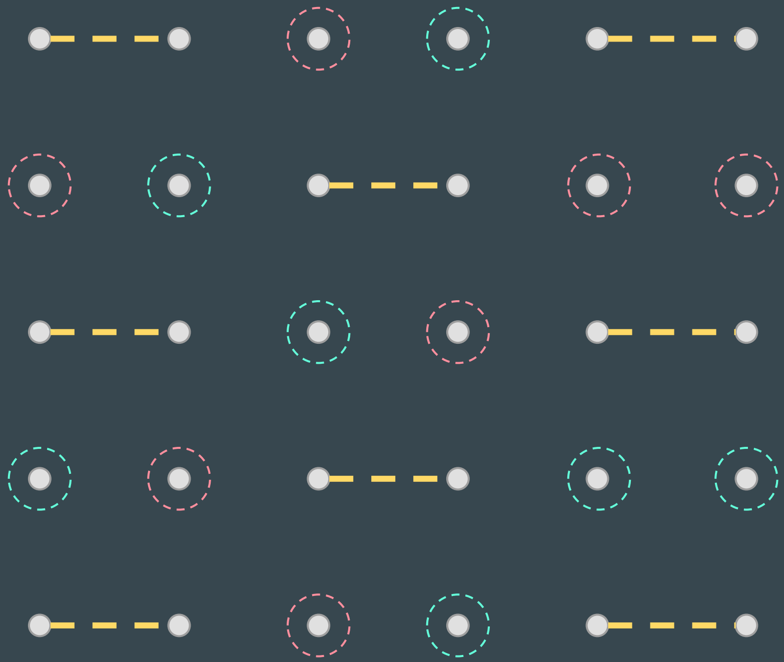
```
Cycle of Hadamard gates
For d clock cycles:
          Apply CZs
          If no CZ applied
                    If no random gate acted yet
                              Apply T
                    Else
                              Apply gate different from previous
```



Random Circuit

For example:

```
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
```



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.

Relational problem which can be verified in classical exponential time by calculating ideal probabilities

Under what assumption is HOG classical hard

Quantum Threshold assumption:

There is no polynomial time classical algorithm that takes a description of a random quantum circuit C, and that guesses whether $|<0^n|C|0^n>|^2$ *is greater or less than the median of the values of* $|<0^n|C|x>|^2$*, with success probability at least* $\frac{1}{2} + \Omega(\frac{1}{2}^n)$ *over the choice of C.*

Quantum Threshold Assumption

- There is simple reduction
  - HOG is not hard ⇒ there exists polynomial-time algorithm to find high probability outputs ⇒ one can use this algorithm to guess $|<0^n|C|0^n>|^2$ ⇒ QUATH does not hold
- Despite similarity between HOG and QUATH, importantly it is not a relational problem and does not refer to sampling.
- Justified through rather flimsy reasoning

How Does This Relate to Our Comments From Before

1. A reason Chad must use a quantum computer
    - If QUATH hold then he’ll have to
2. Property of the outcome, which is “highly correlated” to the outcome, to check
    - Did he meet the conditions of the HOG problem?
3. What price did I pay for removing the backdoor that helps us check property
    - Actually it takes exponential time to check this... You just have to brute force it
4. Means to implement on NISQ devices
    - Random circuits are *THE* NISQ device ... google it

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
    - Output entropy equal to Porter-Thomas distribution
- Zero for uniform distribution

Achiever supremacy in range:

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

A Classical Computer Cannot Pass a Cross-Entropy Test?

Approximating cross entropy difference (probably) requires explicitly calculating probabilities

1. This means C = 0 for large circuit
2. Also means we cannot measure cross entropy difference for large circuits

whispers we can probably just extrapolate *whispers*

It is argued that approximating the probabilities is hard and a weaker assumption than QUATH

How Does This Relate to Our Comments From Before

1. A reason Chad must use a quantum computer
   - Producing Porter-Thomas distributions requires a quantum computer
2. Property of the outcome, which is “highly correlated” to the outcome, to check
   - Can cross-entropy benchmark it
3. What price did I pay for removing the backdoor that helps us check property
   - Actually it takes exponential time to check this... You just have to brute force it
4. Means to implement on NISQ devices
   - Random circuits are *THE* NISQ device ... google it

How Does This Relate to Our Comments From Before

1. A reason Chad must use a quantum computer
   - Producing Porter-Thomas distributions requires a quantum computer
2. Property of the outcome, which is "highly correlated" to the outcome, to check
   - Can cross-entropy benchmark it
3. What price did I pay for removing the backdoor that helps us check property
   - Actually it takes exponential time to check this... You just have to brute force it
4. Means to implement on NISQ devices
   - Random circuits are *THE* NISQ device ... google it

Spoiler! It doesn't work anyway

What Have We Learned

- Hypothesis tests are used to prove “quantumness”
- They require a property which should be checked that is “highly correlated” to the hard problem being implemented
- This highly correlated property is sort of the key here

What Have We Learned

We've learned we should wear sunscreen

- Hypothesis tests are used to prove “quantumness”
- They require a property which should be checked that is “highly correlated” to the hard problem being implemented
- This highly correlated property is sort of the key here

Future Work

- 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 the IQP random circuits be restricted to square lattices nicely
    - Can we combine runtime of IQP into Random circuit NISQness

Building Trust For Quantum States

Quantum State Tomography

- Reconstructing the density matrix of a quantum state (output of an experiment)
- Many measurements and various measurement settings
- Scales exponentially in the number of subsystems (accounting for all correlations)
- *Independently and Identically Distributed* (IID) assumption

Quantum State Certification

Target state ρ, direct fidelity estimation

$$1 - \sqrt{F(\rho,\sigma)} \leq \mathrm{Tr}(|\rho - \sigma|) \leq \sqrt{1 - F(\rho,\sigma)}$$

IID assumption: $\sigma^N = \sigma^{\otimes N}$

Quantum State Verification

Target state ρ,

$F(\rho, \sigma) \geq 1 - \epsilon$ with probability greater than $1 - \delta$

where $N = poly(\frac{1}{\epsilon}, \frac{1}{\delta})$

No IID assumption!

Quantum State Verification Beyond Tomography

Target state ρ,

$F(\rho^{\otimes m}, \sigma^m) \geq 1 - \epsilon$ with probability greater than $1 - \delta$

where $N = poly(m, \frac{1}{\epsilon}, \frac{1}{\delta})$

No IID assumption!

What About CV?

Infinite Fock basis: $\{|n\rangle\}_{n\in\mathbb{N}}$

$$\rho = \sum_{k,l=0}^{\infty} \rho_{kl} |k\rangle \langle l|$$

➔ We are not going to verify all of it: energy cutoff

$$\rho \approx \sum_{k,l=0}^{E} \rho_{kl} |k\rangle \langle l|$$

CV Quantum State Tomography

- Finite support over the Fock basis assumption

- IID assumption

Estimating σ_{kl} for $k, l \leq E$

$$\sigma_{\leq E}^{\otimes N}$$

CV Quantum State Certification

- Energy test

- IID assumption

Estimating $F(\rho, \sigma)$ or $\mathrm{Tr}(A\sigma)$ efficiently

$$\sigma^{\otimes N}$$

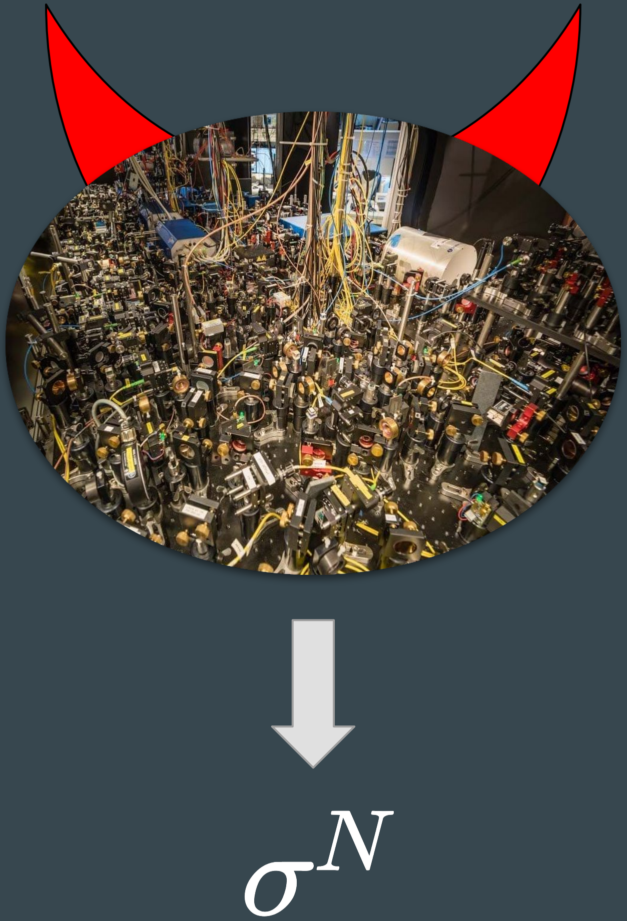
CV Quantum State Verification

- Refined energy test

- No assumptions

Estimating $F(\rho^{\otimes m}, \sigma^m)$ efficiently

(Proof using De Finetti theorem)



Outlook

- Extending crypto techniques to CV (no obvious twirling lemma)
- More flexible definitions of security: different measures, robust definitions
- Tailored protocols: trading efficiency and security

Thanks!