Bayesian Estimation of Error Models for Improving Circuit Compilation

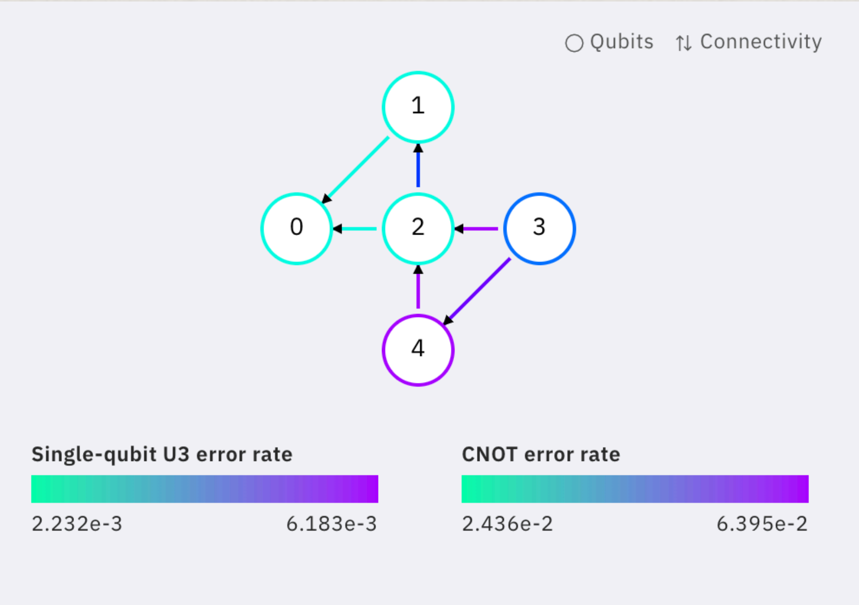
William S. Moses

wmoses@mit.edu

August 1, 2019

Quantum Circuits

- Functional near-term quantum computing will require learning how to make effective use of noisy error-prone hardware.
- Makes things difficult for quantum algorithm designers as they must simultaneously deeply understand the hardware and how to best map to it.
- Practically, often means "try to use the less noisy qubits more"



Error-Based Synthesis

- There are many types of potential quantum errors
  - (P.S. Hardware folks please tell me what you see happens in practice.)
- We can understand errors as modifications to gates in our original circuit
  - Rx(theta) => Rx(theta) U(phi)
  - CNOT => if rand() < .2 then CNOT CZ else CNOT
  - U => u ~ Distribution(U)

Idea: Synthesize With Errors

- When synthesizing a desired circuit to run on hardware, consider build with error-prone rather than theoretical gates
- Synthesis should find the physical circuit that closest matches the desired circuit in expectation (over gate error models)
- Should be able to estimate the error of the circuit ahead-of-time
- Let's see some examples!

Simple Example

- We found that running a theoretical rotation gate on the hardware results in an overrotation of 0.1

Desired Circuit:

$$\boxed{Rx(\pi)}^{T}$$

Error Model:

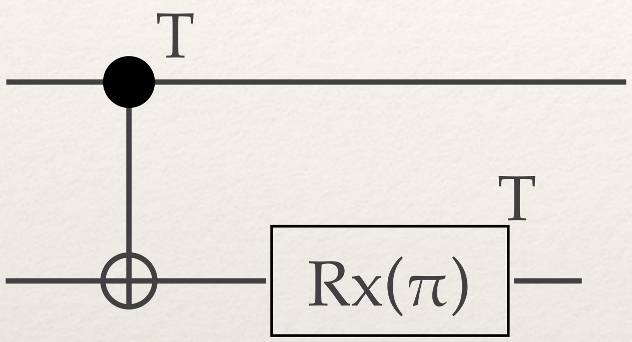
$$\boxed{Rx(\theta)} \rightarrow \boxed{Rx(\theta)}^{T} \boxed{Rx(0.1)}^{T}$$

- We should submit our desired circuit underrotated by 0.1 to compensate for the hardware mapping

$$\boxed{Rx(\pi\text{-}0.1)} \rightarrow \boxed{Rx(\pi\text{-}0.1)}^{T} \boxed{Rx(0.1)}^{T} = \boxed{Rx(\pi)}^{T}$$

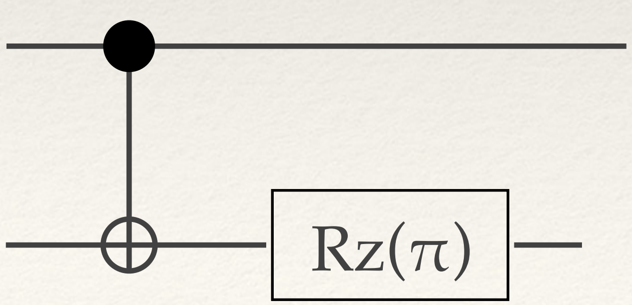
More Complex Example

Desired Circuit:

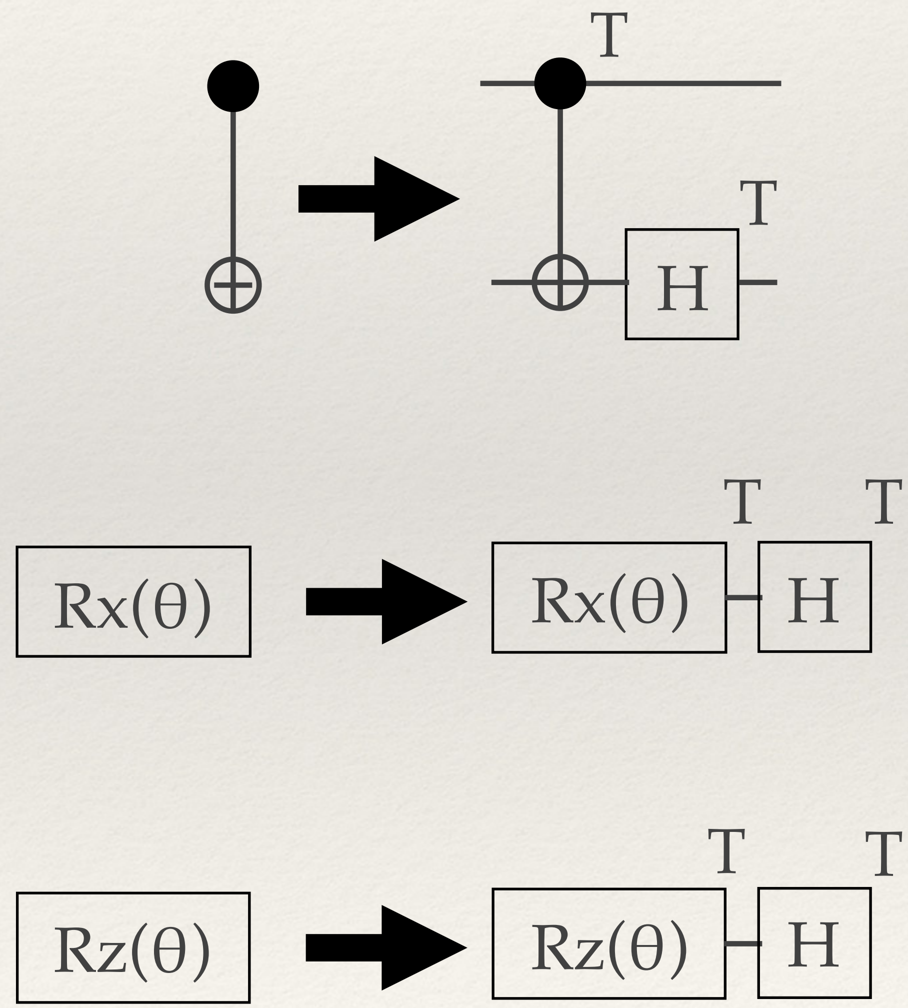


≈

Submitted Circuit



Error Model:



Expectation Example

Desired Circuit:

$$\boxed{Rx(\theta)}^T$$

Error Model:

$$\boxed{Rx(\theta)} \rightarrow \begin{cases} \boxed{Rx(\theta)}^T & \text{if rand()} < \frac{7}{10} \\ \boxed{1}^T & \text{else} \end{cases}$$

$$\boxed{Rz(\theta)} \rightarrow \begin{cases} \boxed{Rz(\theta)}^T & \text{if rand()} < \frac{99}{100} \\ \boxed{1}^T & \text{else} \end{cases}$$

$$\boxed{H} \rightarrow \begin{cases} \boxed{H}^T & \text{if rand()} < \frac{99}{100} \\ \boxed{1}^T & \text{else} \end{cases}$$

Expectation Example

Submitted Circuit

H — Rz(θ) — H

97%
chance of exactly correct

Suboptimal Circuit

Rx(θ)

70%
chance of exactly correct

Importance of Error Modeling

- Error-corrective synthesis ***requires*** good approximations of gate error behavior to produce reasonable circuits / estimates
- This requires the ability to estimate the error distributions

A Puzzle

- Which circuit produced these results?

$$600 \quad |0\rangle \qquad\qquad 400 \quad |1\rangle$$

$$\text{a)} \quad |0\rangle - \begin{cases} \boxed{X} & \text{if rand()} < \frac{99}{100} \\ \boxed{1} & \text{else} \end{cases} - \boxed{\text{Measure(Z)}}$$

$$\text{b)} \quad |0\rangle - \begin{cases} \boxed{H} & \text{if rand()} < \frac{99}{100} \\ \boxed{1} & \text{else} \end{cases} - \boxed{\text{Measure(Z)}}$$

A Puzzle

$600 \quad |0\rangle \qquad\qquad 400 \quad |1\rangle$

a) $$|0\rangle - \begin{cases} \boxed{X} & \text{if rand()} < \frac{99}{100} \\ \boxed{1} & \text{else} \end{cases} - \boxed{\text{Measure(Z)}}$$

$1\% \quad |0\rangle \qquad\qquad 99\% \quad |1\rangle$

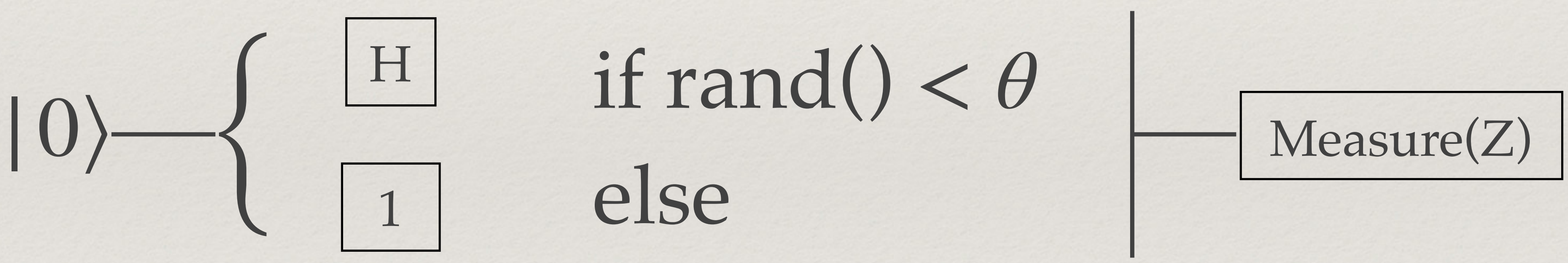
b) $$|0\rangle - \begin{cases} \boxed{H} & \text{if rand()} < \frac{99}{100} \\ \boxed{1} & \text{else} \end{cases} - \boxed{\text{Measure(Z)}}$$

$50.5\% \quad |0\rangle \qquad\qquad 49.5\% \quad |1\rangle$

A Slightly Harder Puzzle

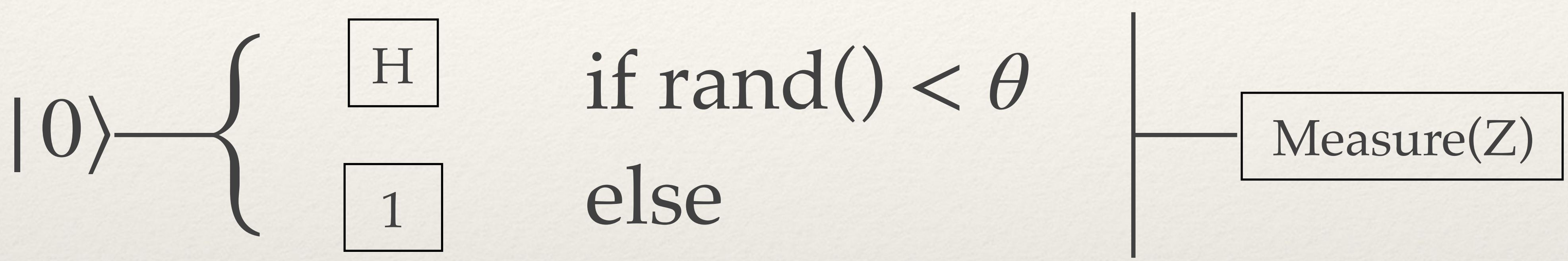
- Which parameter θ produced these results?

600 $|0\rangle$ 400 $|1\rangle$



A Slightly Harder Puzzle

- Which parameter θ produced these results?



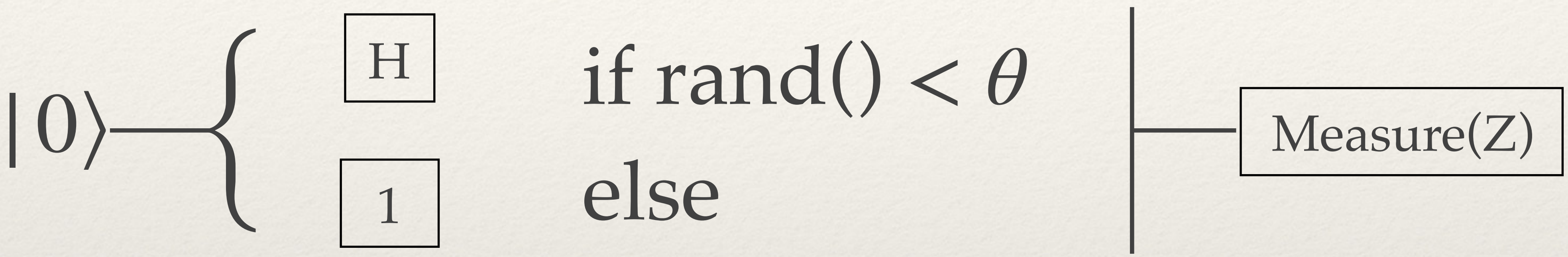
$$p(data\,|\,\theta) = \binom{1000}{\text{data}} \left(\frac{\theta}{2} + (1-\theta)\right)^{\text{data}} \left(\frac{\theta}{2}\right)^{\text{1000-data}}$$

$$p(\theta\,|\,data) = \frac{p(data\,|\,\theta)p(\theta)}{p(data)}$$

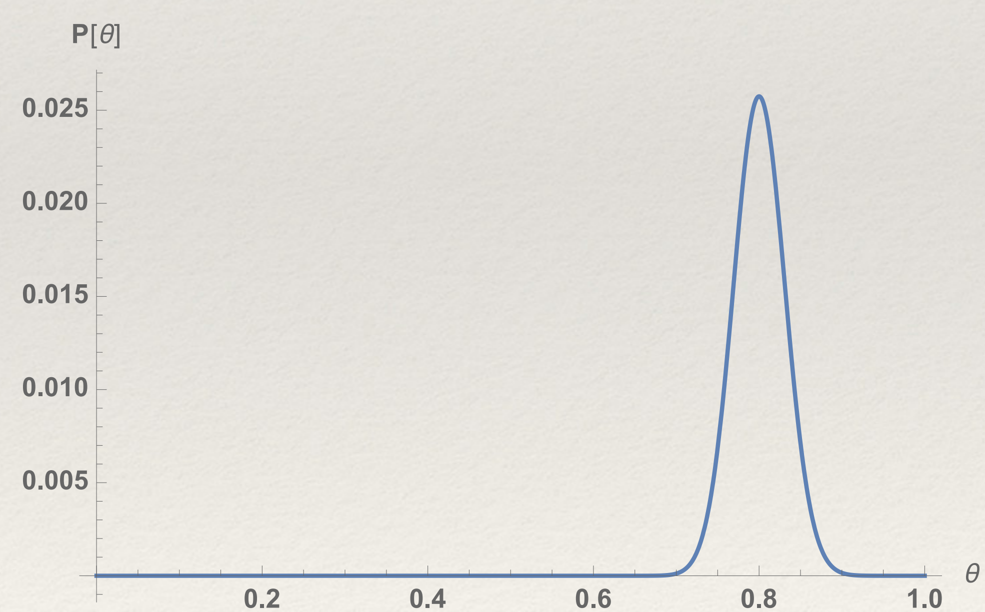
$$\theta^* = \arg\max_{\theta} p(data\,|\,\theta)p(\theta)$$

A Slightly Harder Puzzle

- Which parameter θ produced these results?



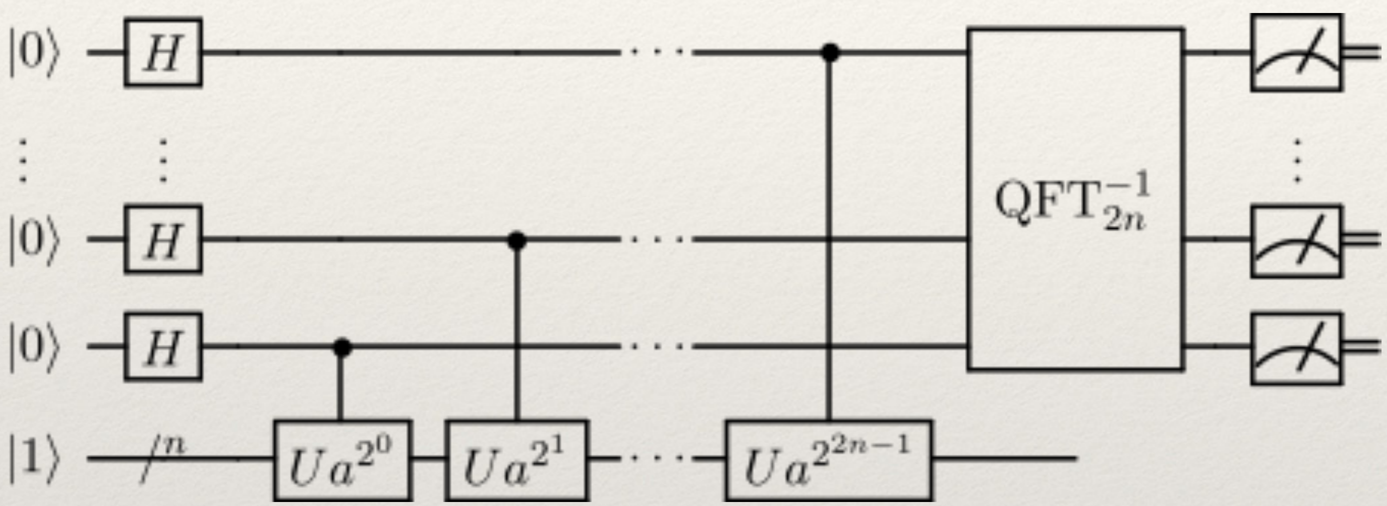
$$\theta^* = \arg\max_{\theta} p(data\,|\,\theta)p(\theta)$$



$$\theta^* = 0.8$$

A Much Harder Puzzle

- Which parameters θ, ϕ produced these results?



$$\boxed{\mathrm{U}} \rightarrow \begin{cases} \boxed{\mathrm{U}} & \text{if rand()} < \theta \\ \boxed{1} & \text{else} \end{cases}$$

$$\boxed{\mathrm{H}} \rightarrow \begin{cases} \boxed{\mathrm{H}} & \text{if rand()} < \phi \\ \boxed{1} & \text{else} \end{cases}$$

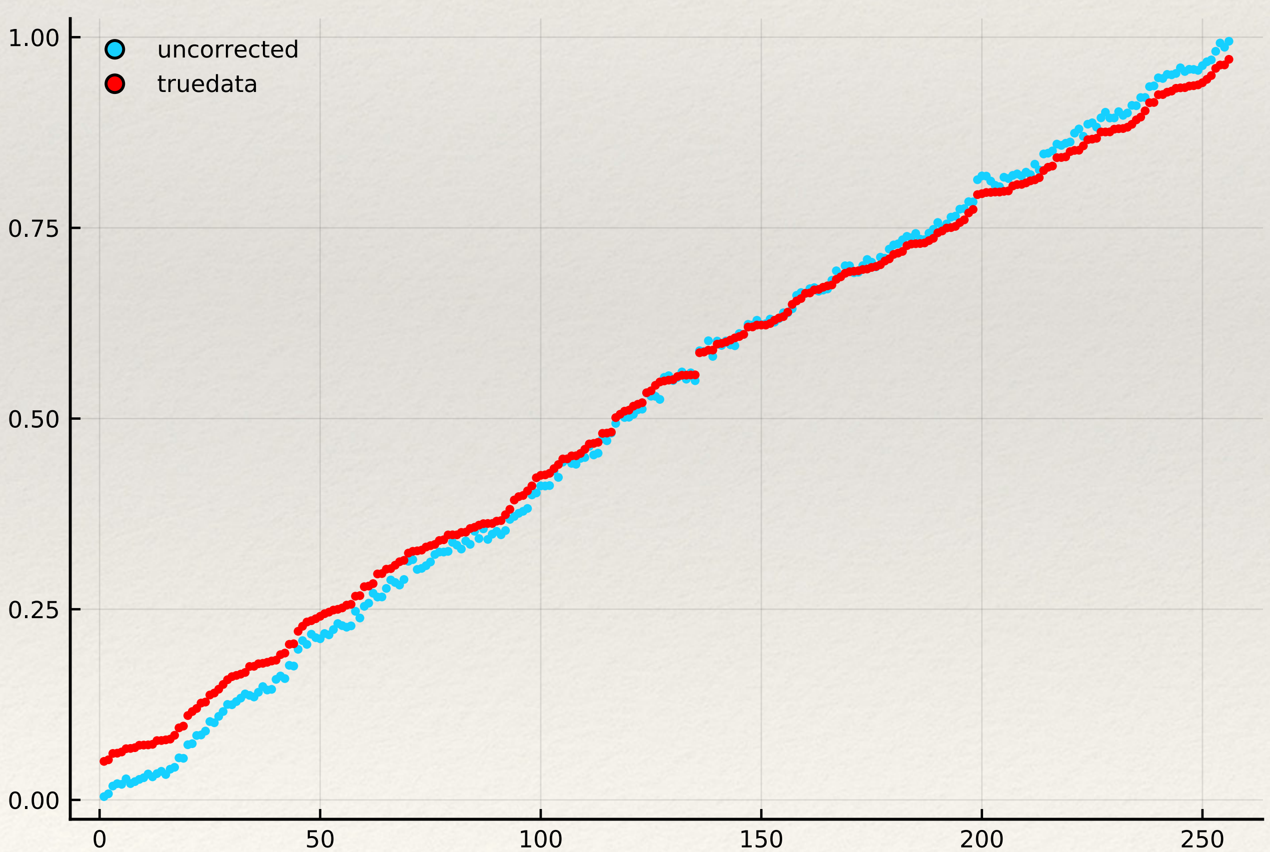
Methodology

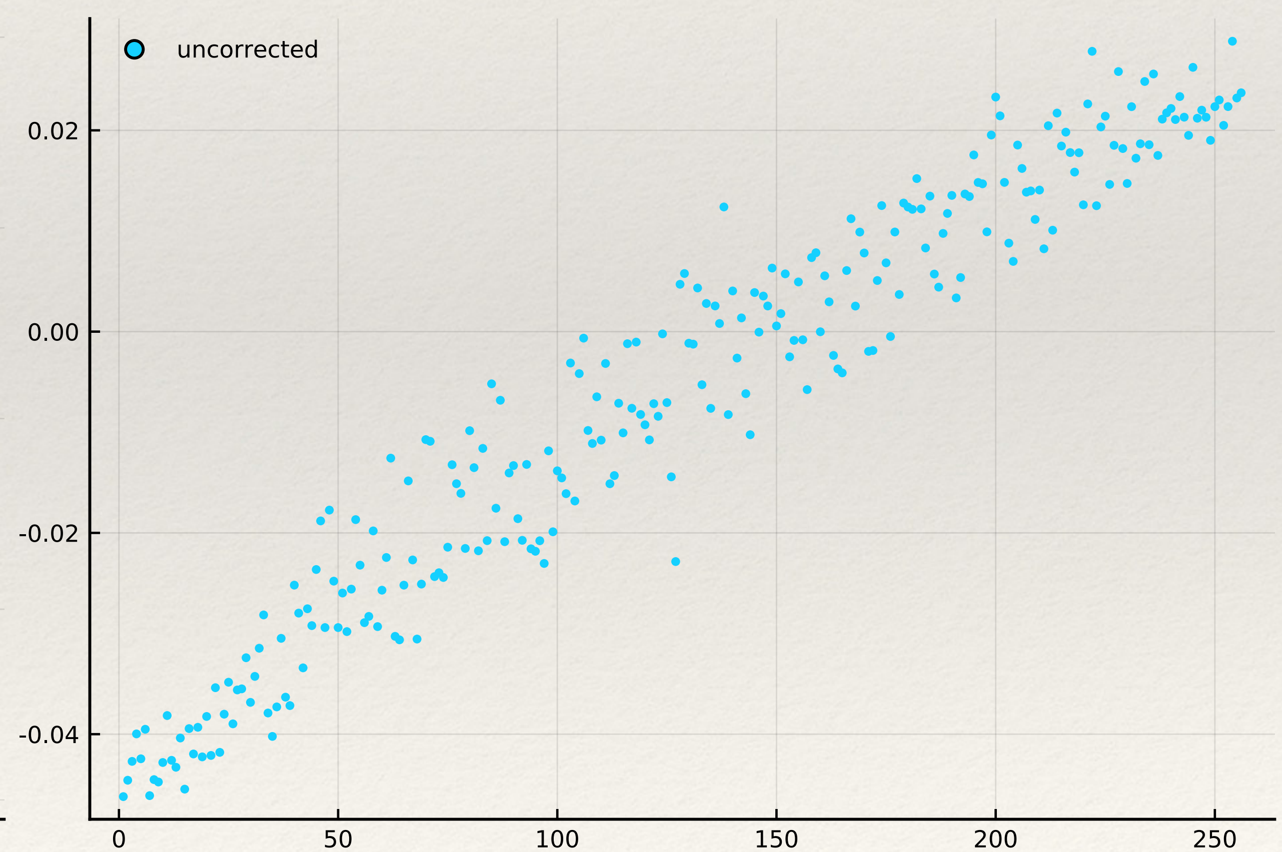
- Run a number of unaltered circuits on hardware
- Build a version of the circuits using a parameterized error model and solve for the most likely parameters
- Use those parameterized error models to synthesize better circuits
- (Optional) repeat using that new data to find better error models

Case Study: Random Unitary

$$\boxed{U3(\theta, \phi, \lambda)} = \boxed{Rz(\phi+3\pi)} - \boxed{Rx(\pi/2)} - \boxed{Rz(\theta+3\pi)} - \boxed{Rx(\pi/2)} - \boxed{Rz(\lambda)}$$

Sample many θ, ϕ, λ and run on IBM quantum hardware



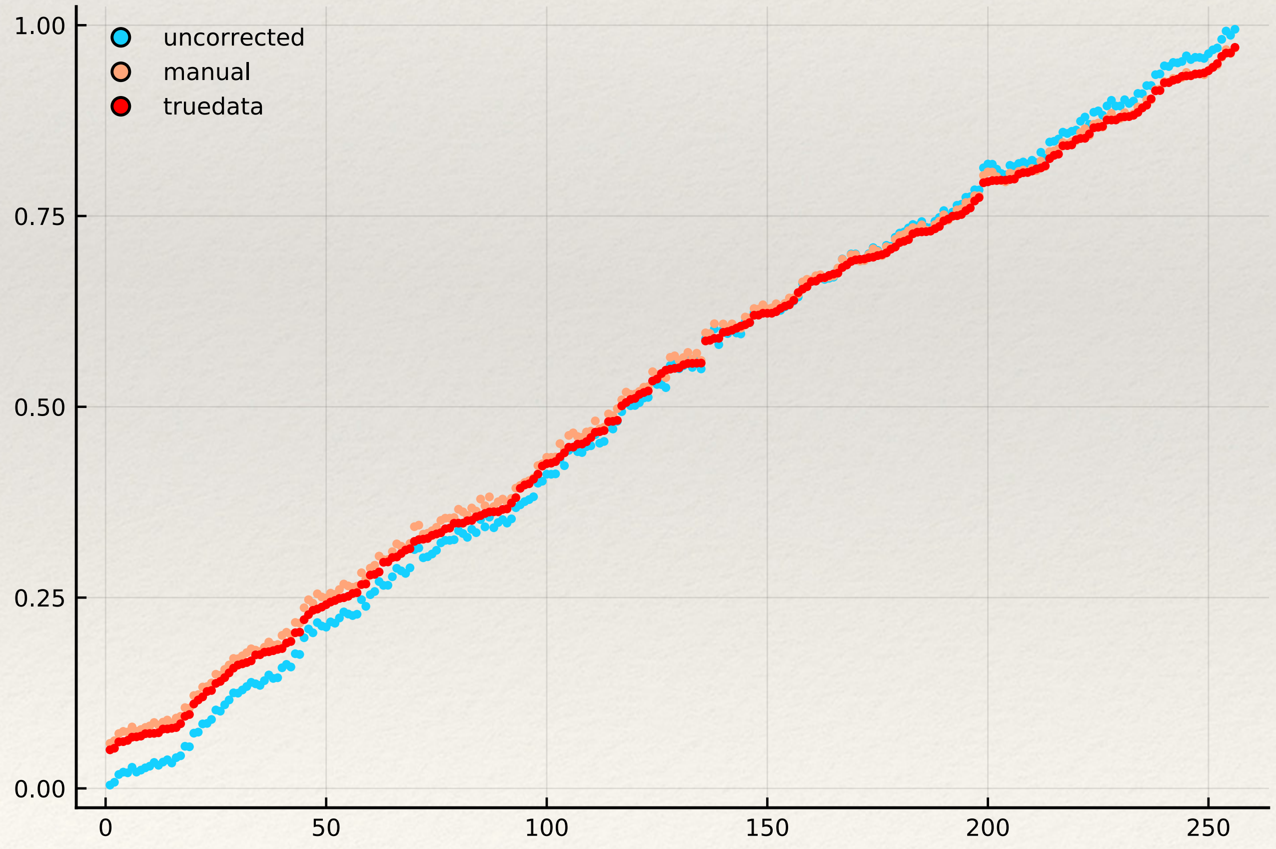


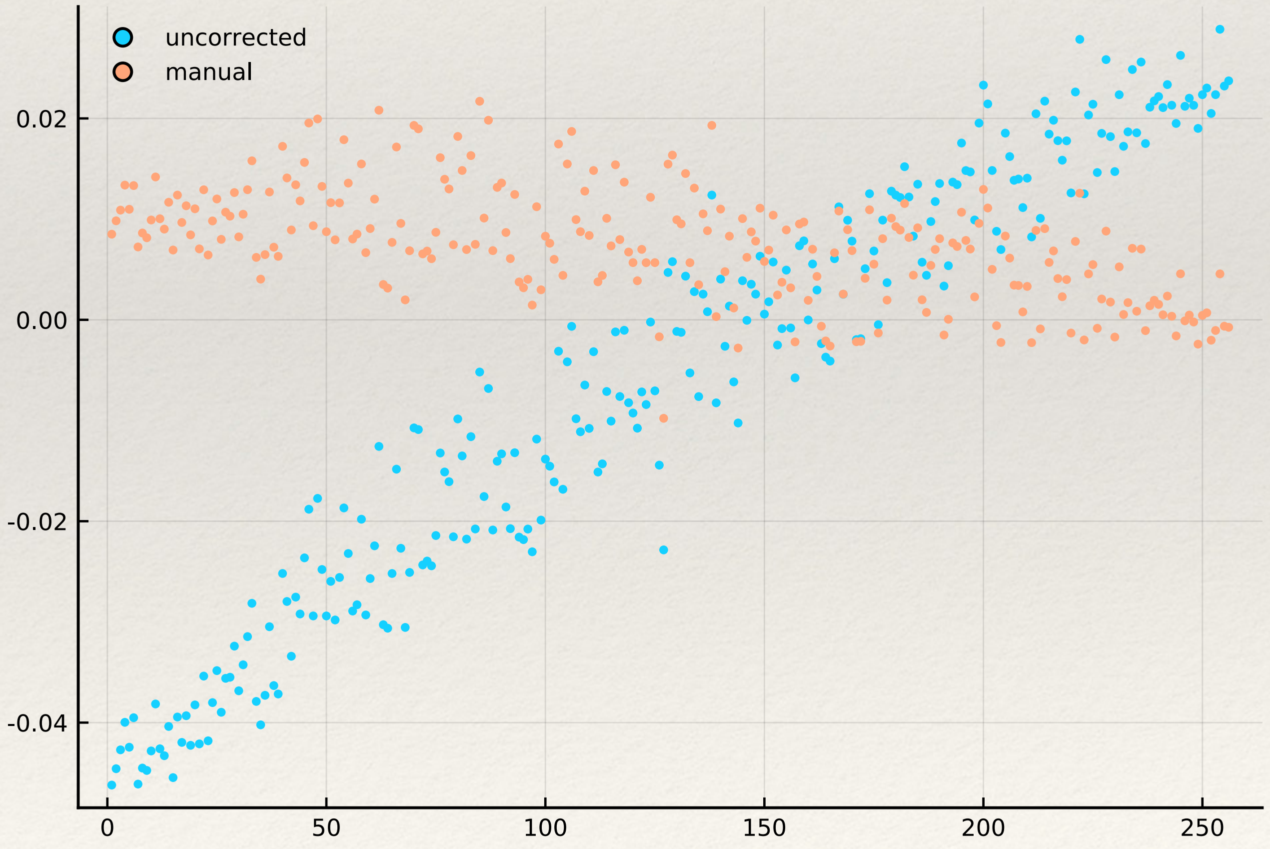
Case Study: Random Unitary

$|0\rangle$ — Measure(Z)

$|0\rangle$ — X — Measure(Z)

Manually estimate measurement error using above experiment





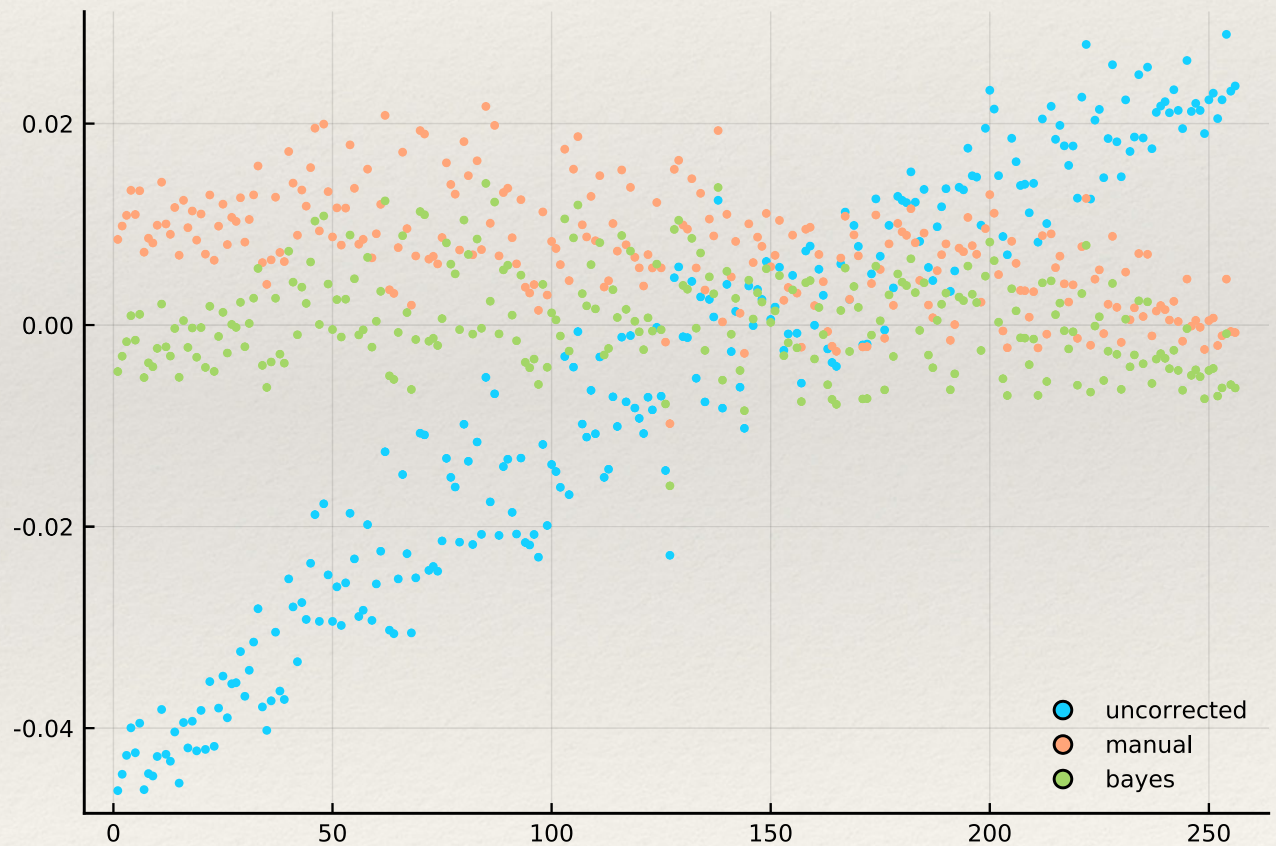
Case Study: Random Unitary

Instead: derive most probable error rates from data

```
@gen function scopeU3(scope)
    θ = @trace(uniform(0, 2*pi), (scope,:θ))
    φ = @trace(uniform(0, 2*pi), (scope,:φ))
    λ = @trace(uniform(0, 2*pi), (scope,:λ))
    u3(θ, φ, λ)
end

@gen function vectormaker(samples::Int, points::Int)
    data = Float64[];
    zeroToOne = @trace(Gen.beta(2, 5), (:measure, :zeroToOne))
    oneToZero = @trace(Gen.beta(2, 5), (:measure, :oneToZero))
    for i in 1:points
        u1 = @trace(scopeU3(i))
        state = zero_state(1)
        prob0 = probs(Yao.apply!(state, u1))[1]
        prob2 = prob0 * (1-zeroToOne) + (1-prob0) * oneToZero
        z0 = @trace(binomial(prob2, samples), (i, :z0))
        push!(data, prob2)
    end
    data
end

observations = Gen.choicemap()
for i in 1:size(data, 1)
    observations[(i, :θ)] = data[i,1]
    observations[(i, :φ)] = data[i,2]
    observations[(i, :λ)] = data[i,3]
    observations[(i, :z0)] = datag[i,1]
end
```

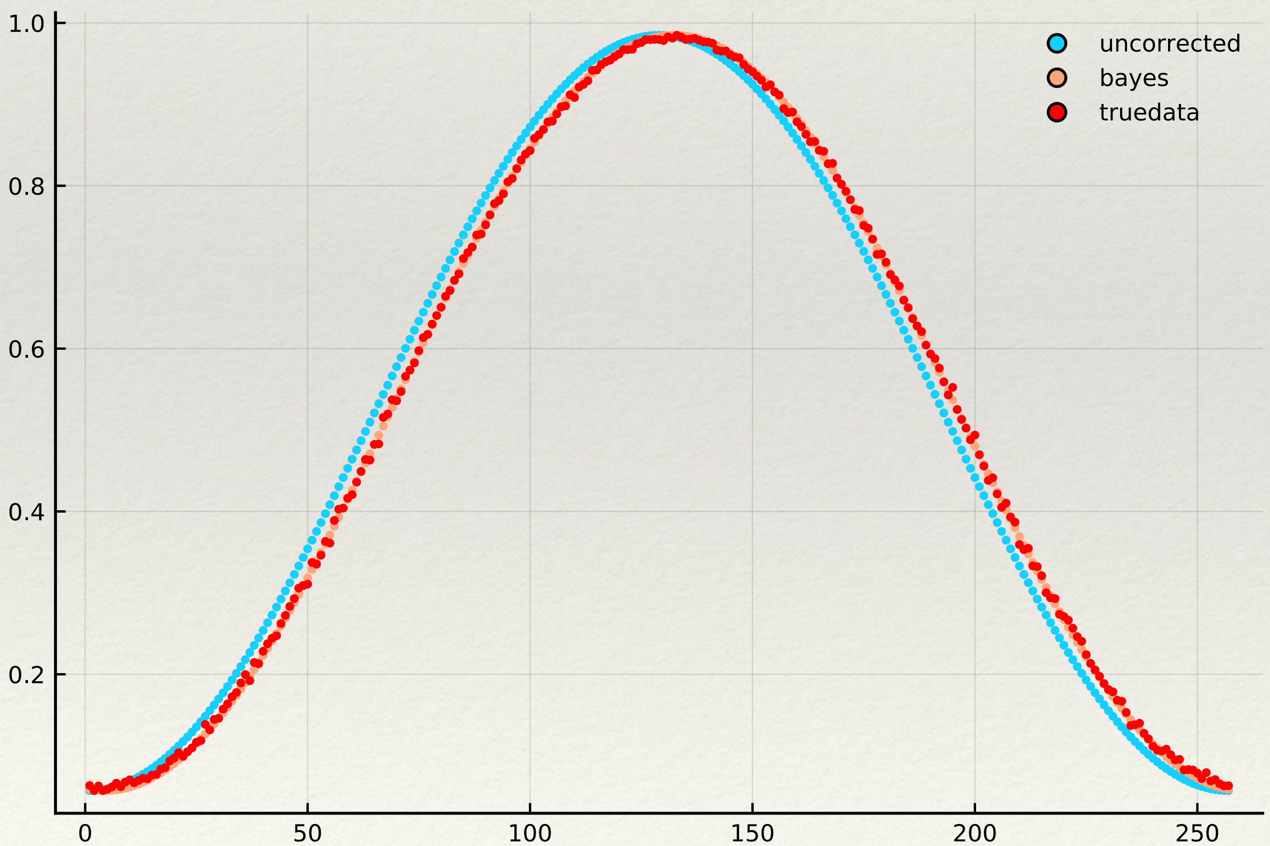


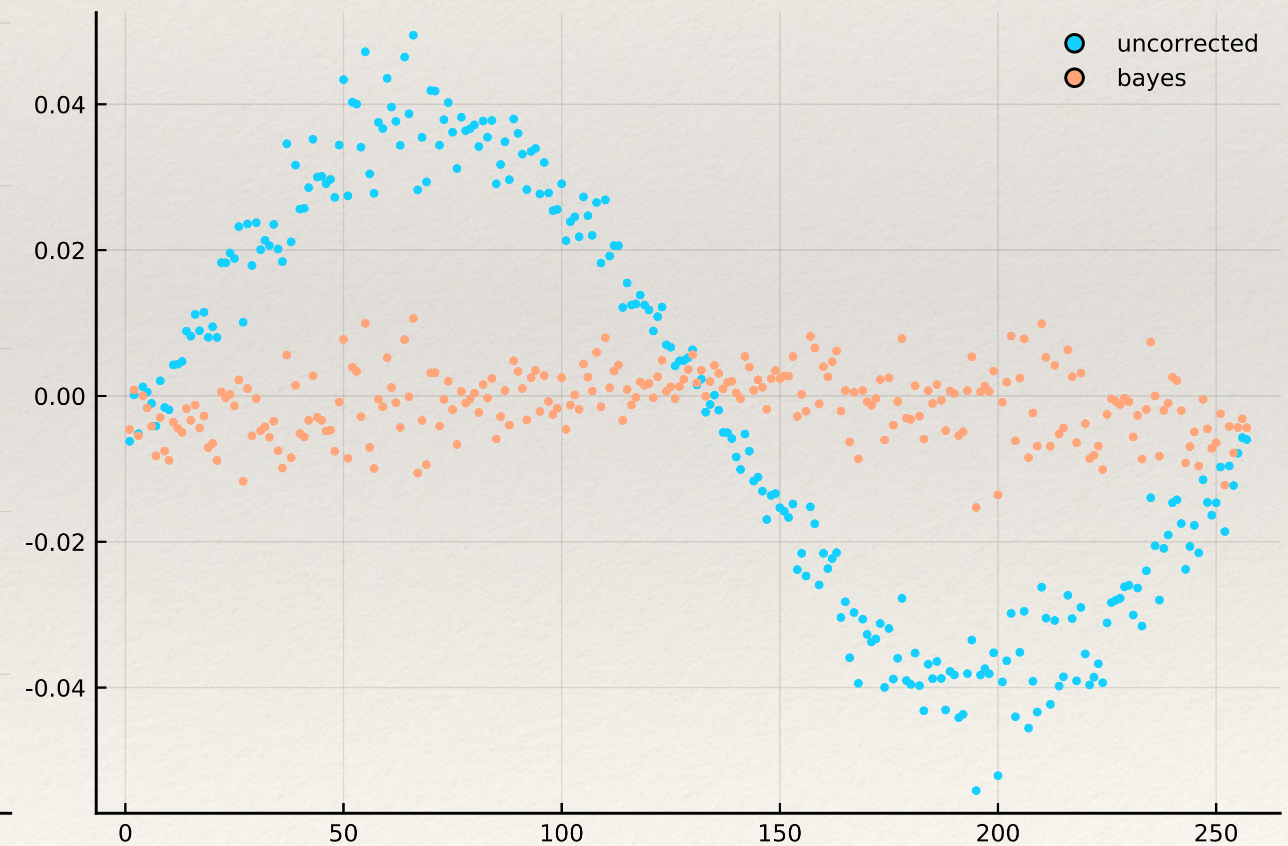

```
(trace, lml_est) = Gen.importance_resampling(vectormaker, (datag[1,1]+datag[1,2],size(data, 1)), observations, 1);
```

Case Study: Shifted Unitary

U3(θ, ϕ, λ) = Rz(ϕ+3π) — Rx(π/2) — Rz(θ+3π) — Rx(π/2) — Rz(λ)

Sample θ and run on IBM quantum hardware, found shift





Future Directions

- More experiments!
  - Derive over-correction parameters
  - Derive std dev of errors
  - Derive random Pauli probability