

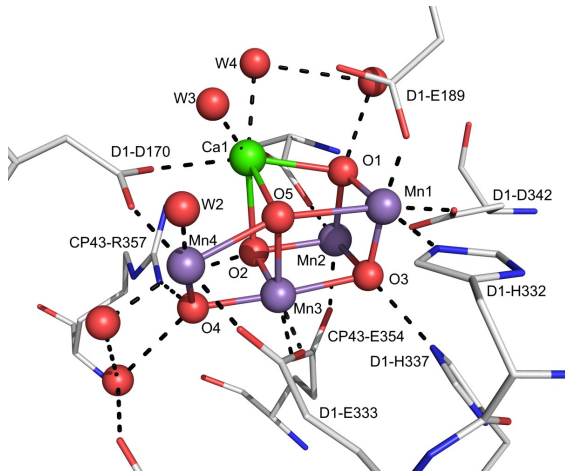


Dissipation Nation

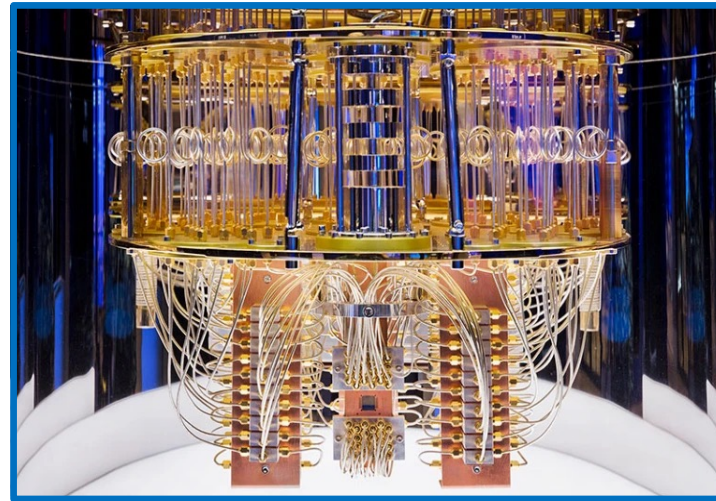
Sabee Grewal, Matthew Hagan, Daniel
Marti-Dafcik, Lucia Vilchez-Estevez

What is the Dissipative Quantum Eigensolver?

Algorithm to prepare a ground state due to T. Cubitt (arxiv:2303.11962)



Nature. 473 (7345): 55–60



Nature 599, 542 (2021)



ρ

What is the Dissipative Quantum Eigensolver?

```
 $\rho$  = initial state
until (stop condition):
     $\rho, b = E(\rho)$ 
    if b:
        Do some qubit resets
return  $\rho$ 
```

ρ will be close to the true ground state!

What is the Dissipative Quantum Eigensolver?

```
 $\rho$  = initial state  
until (stop condition):  
     $\rho, b = E(\rho)$   
    if b:  
        Do some qubit resets  
return  $\rho$ 
```



Approximate GS projection

ρ will be close to the true ground state!

Why DQE?

Scientifically important

- Ground state preparation central in quantum chemistry and materials!
- Enables computing important properties
- Example: energy \rightarrow reaction rates

Advantages of DQE

- Provable guarantees
- Noise resilient
- Conceptually simple

Why DQE? A good fit for Triple Alpha!

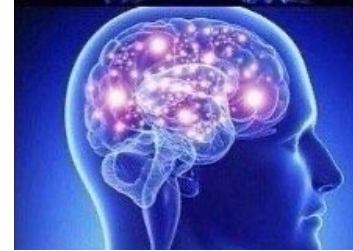
```
ρ = initial state
until (stop condition):
    ρ, b = E(ρ)
    if b:
        Do some qubit resets
return ρ
```



Why DQE? A good fit for Triple Alpha!

```
 $\rho$  = initial state  
until (stop condition):  
     $\rho, b = E(\rho)$   
    if b:  
        Do some qubit resets  
return  $\rho$ 
```

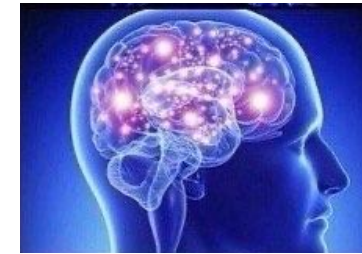
Looping uses the Turing-completeness
of Helium



Why DQE? A good fit for Triple Alpha!

```
ρ = initial state
until (stop condition):
    ρ, b = E(ρ)
    if b:
        Do some qubit resets
return ρ
```

Looping uses the Turing-completeness of Helium



Intermediate measurements!
Complicated control of the quantum device made easy.



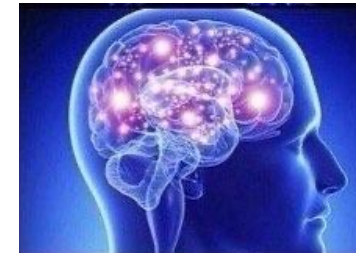
Why DQE? A good fit for Triple Alpha!

```
 $\rho$  = initial state  
until (stop condition):  
     $\rho, b = \mathbf{E}(\rho, \epsilon)$   
    if b:  
        Do some qubit resets  
return  $\rho$ 
```

Interesting parameters to play with! Implementation has practical importance!



Looping uses the Turing-completeness of Helium



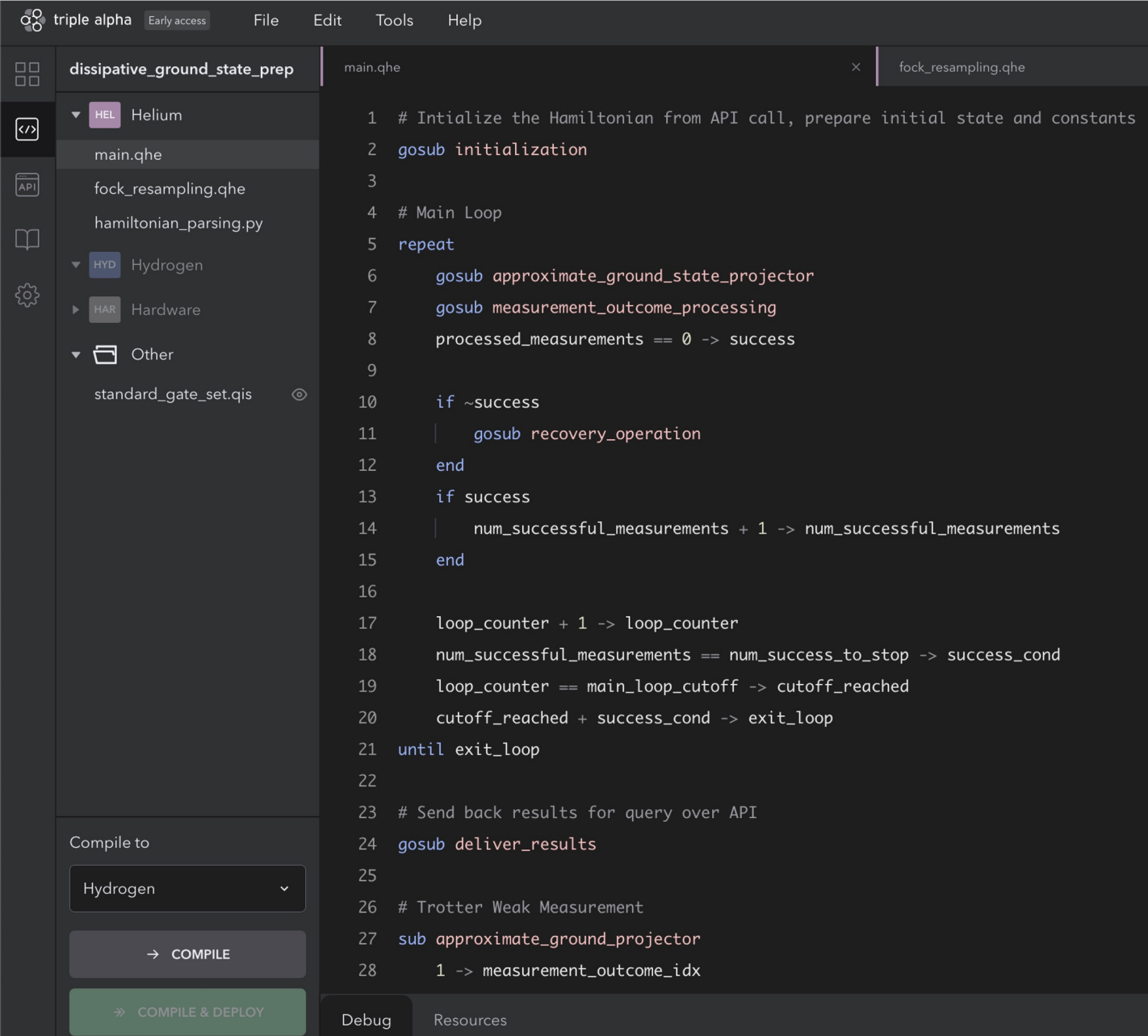
Intermediate measurements!
Complicated control of the quantum device made easy.



What did we do?

~ 200 lines of Helium code!
And Python client code!

Deployed as an API!



```
triple alpha Early access File Edit Tools Help

dissipative_ground_state_prep main.qhe fock_resampling.qhe

HEL Helium
main.qhe
API fock_resampling.qhe
hamiltonian_parsing.py
HYD Hydrogen
HAR Hardware
Other
standard_gate_set.qis

Compile to
Hydrogen
→ COMPILE
» COMPILE & DEPLOY

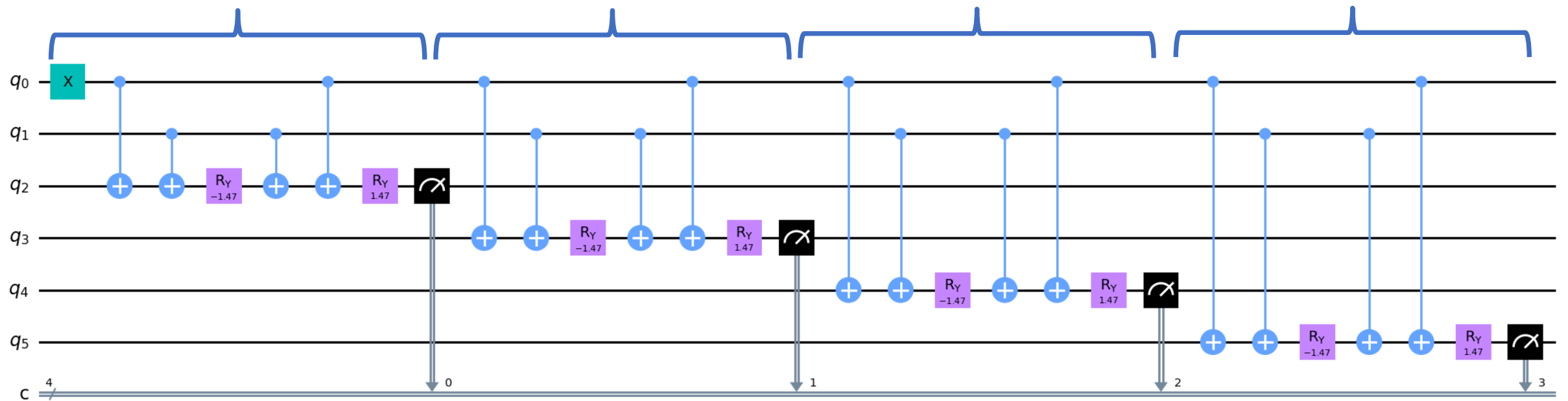
Debug Resources

1 # Initialize the Hamiltonian from API call, prepare initial state and constants
2 gsub initialization
3
4 # Main Loop
5 repeat
6   gsub approximate_ground_state_projector
7   gsub measurement_outcome_processing
8   processed_measurements == 0 -> success
9
10  if ~success
11    | gsub recovery_operation
12  end
13  if success
14    | num_successful_measurements + 1 -> num_successful_measurements
15  end
16
17  loop_counter + 1 -> loop_counter
18  num_successful_measurements == num_success_to_stop -> success_cond
19  loop_counter == main_loop_cutoff -> cutoff_reached
20  cutoff_reached + success_cond -> exit_loop
21 until exit_loop
22
23 # Send back results for query over API
24 gsub deliver_results
25
26 # Trotter Weak Measurement
27 sub approximate_ground_projector
28   1 -> measurement_outcome_idx
```

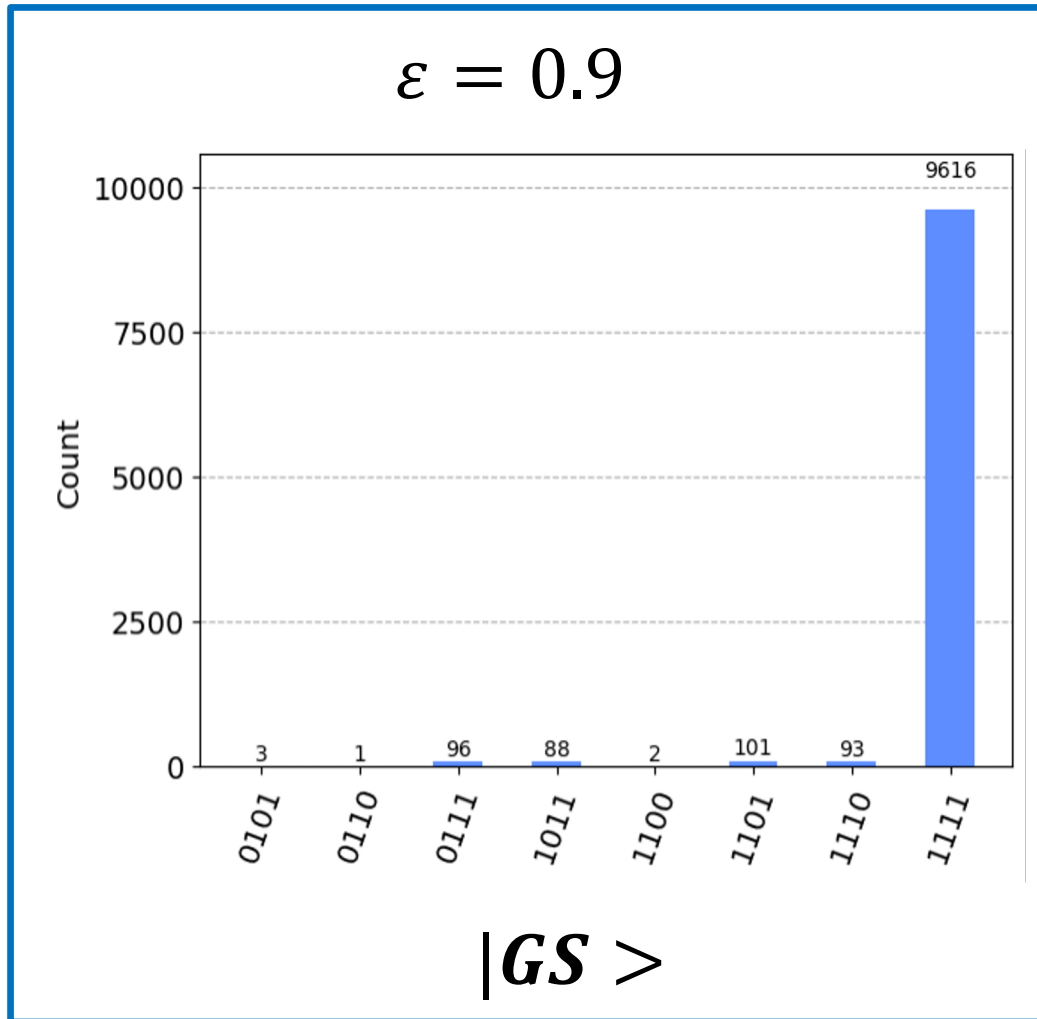
What did we do?

Built down to IBM circuit :)

$H \rightarrow U = \text{approximate GS projector}$

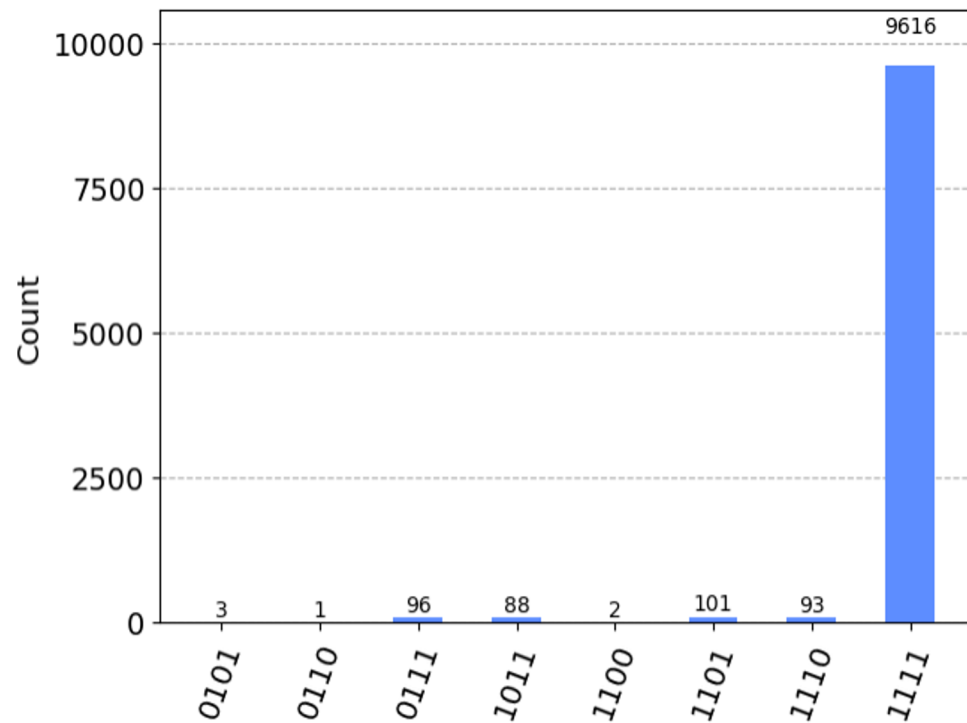


What did we do?



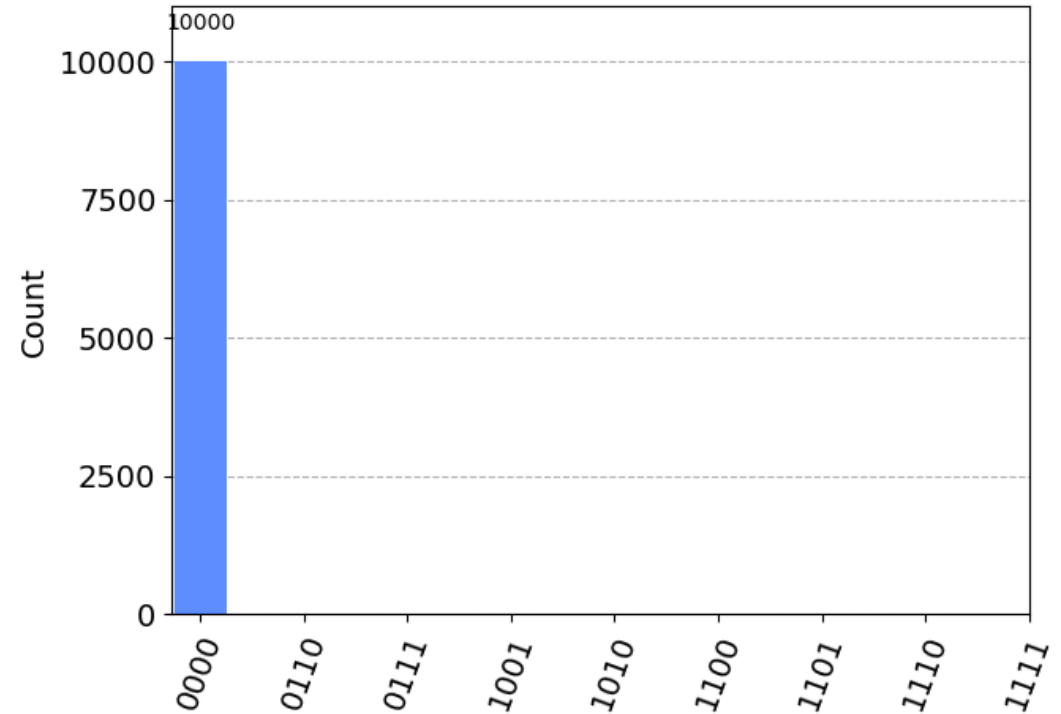
What did we do?

$\varepsilon = 0.9$



$|GS\rangle$

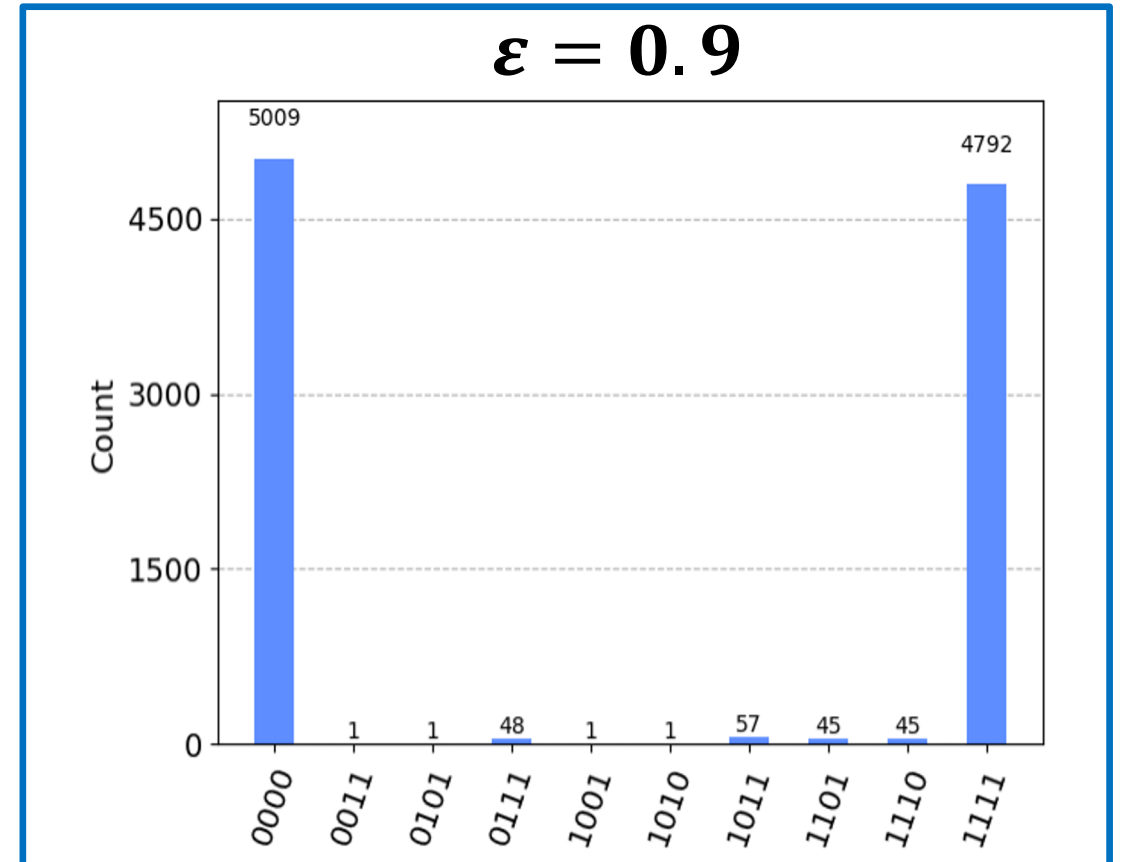
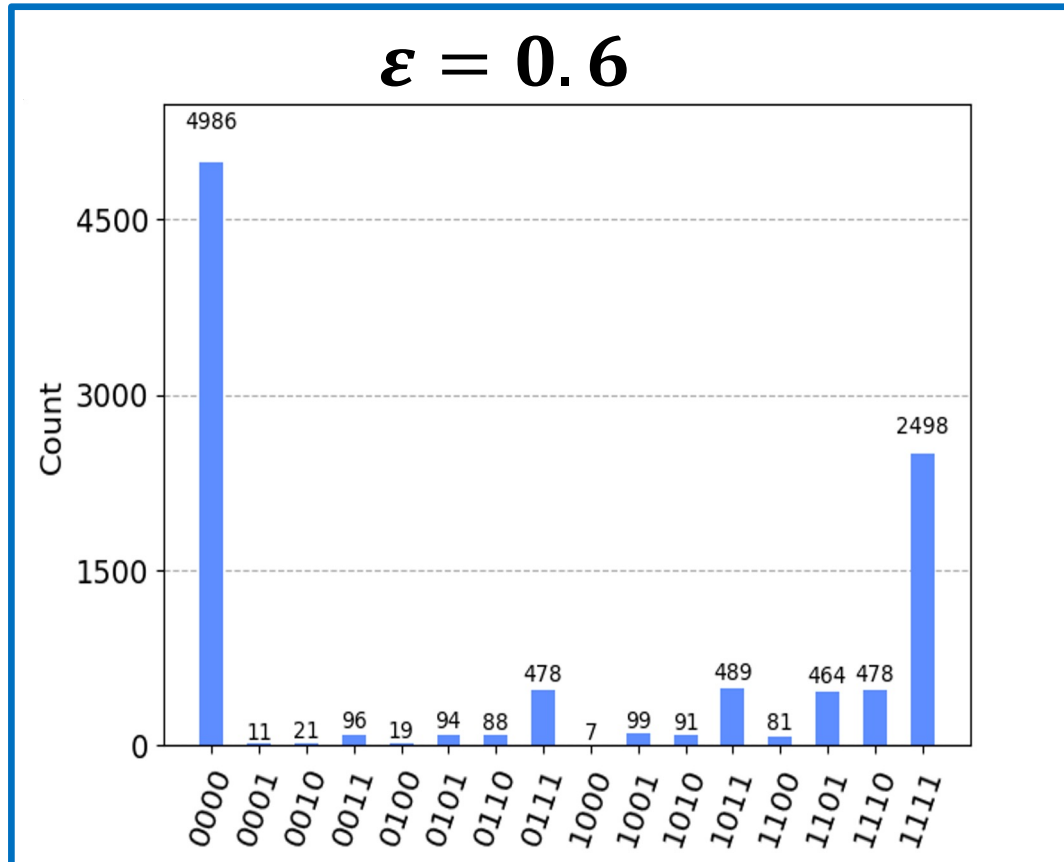
$\varepsilon = 0.9$



$|GS^\perp\rangle$

What did we do?

$$\frac{1}{\sqrt{2}}(|GS\rangle + |GS^\perp\rangle)$$



Decreasing ε makes the projector more gentle

Improvements to Triple Alpha

- Nondemolition measurements
- Client libraries that abstract away curl request.
- More expressive language features:

```
if expression rather than if int  
increment i from A to B
```

- Local IDE and local compiler

Future Work (at Bali next year?!)

- Simulate DQE on H_n chain
- Different instantiations of DQE:
 - Dependence on initial state
 - Resampling
 - Parameter scheduling

Future Work (at Bali next year?!)

- Simulate DQE on H_n chain
- Different instantiations of DQE:
 - Dependence on initial state
 - Resampling
 - Parameter scheduling



Thank you, Horizon,
for the amazing experience!