Solving Eternity II puzzle

by quantum reinforcement learning inspired approach

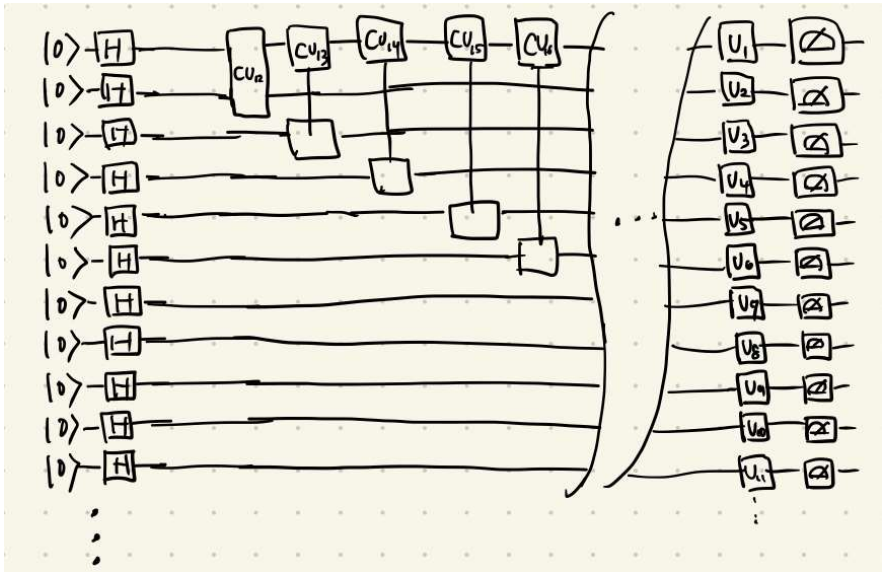
- Edge-matching tiling puzzle
- $2million prize
- Nobody ever solved
- Designed to be difficult to solve by traditional computers

<u>So why don’t we try solving it with a Quantum computer?</u>

Our approach

we had some ideas to solve this puzzle (AS CSP by QAOA…)

We wanted to try quantum reinforcement learning inspired approach.



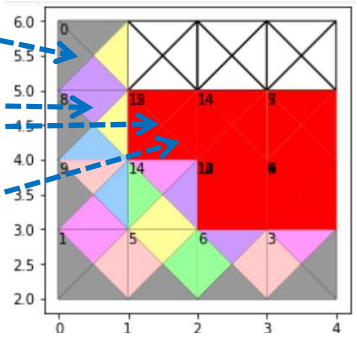
Generate sample

1,
0,
0,
1,
1,
0,
0,
0,
1
.
.
.
.

The result represents the location and the rotation of each puzzle piece

Like if qubit1=0, it means locate the peace 1 to the 2nd place with this rotation…

Simulate the puzzle



Culculate reward according to the level of the completion

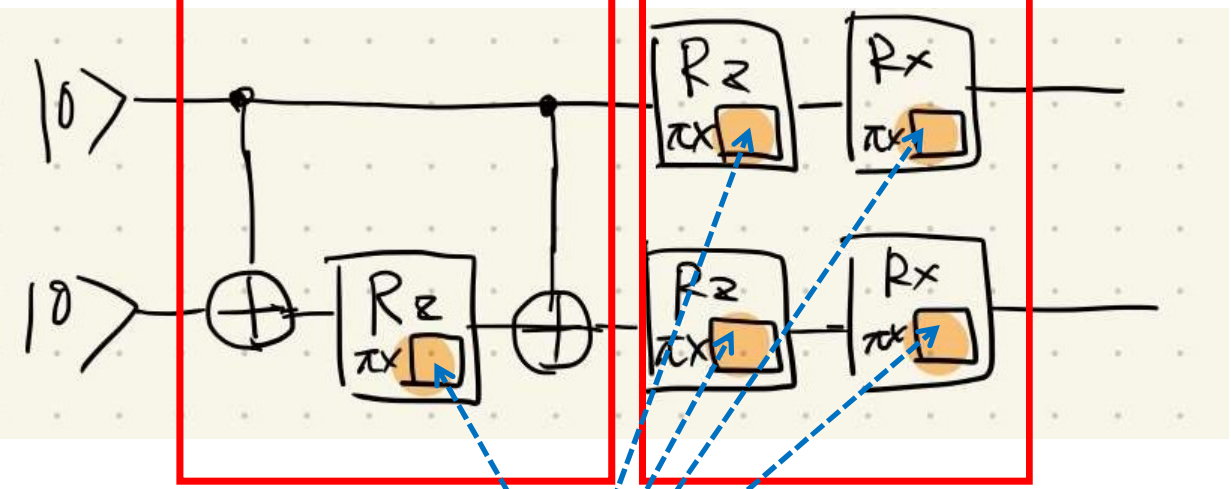
Adjust the circuit

Iterate the process

Basic unit of two qubits

Connection between the two qubits

Coefficients for each qubit



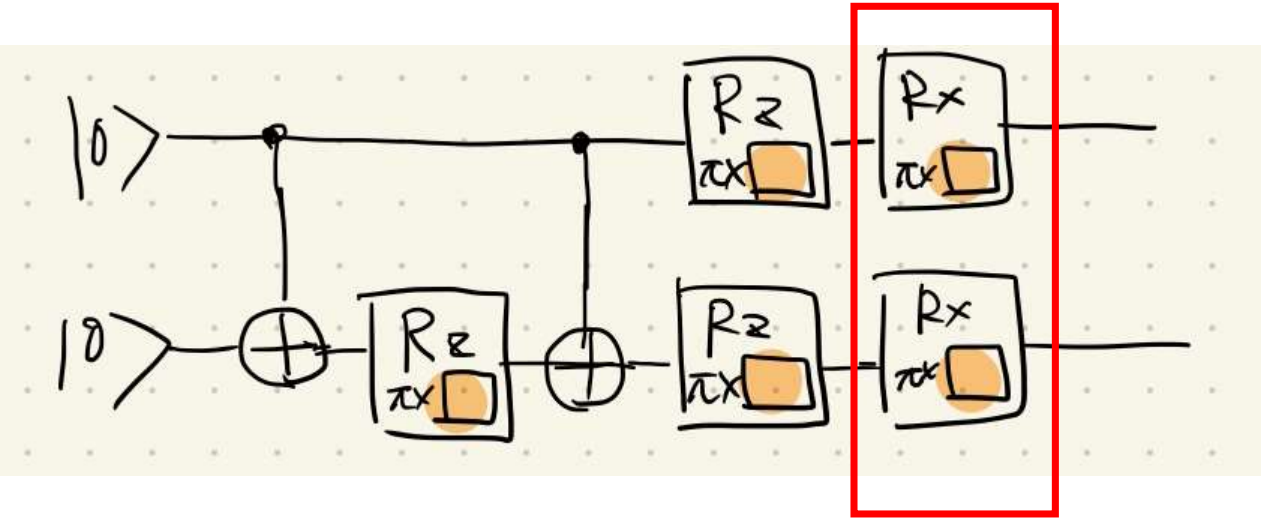
Adjust the rotation of the gates

Calculate the gradient according to the reward

$$
\begin{aligned}
&-\frac{\pi e^{\pi \operatorname{im}(s1)} e^{\pi \operatorname{im}(s2)} e^{\pi \operatorname{im}(s3)} \left|\sin^2\left(\frac{\pi s4}{2}\right)\right|^2 \frac{d}{ds2}\operatorname{im}(s_2)}{4} + \frac{e^{\pi \operatorname{im}(s1)} \frac{\partial}{\partial s2}\operatorname{re}\left(e^{-\frac{i\pi s2}{2}} e^{-\frac{i\pi s3}{2}} e^{-\frac{i\pi \overline{s2}}{2}} e^{-\frac{i\pi \overline{s3}}{2}} \sin^2\left(\frac{\pi s4}{2}\right)\right.}{4} \\
&+\frac{e^{\pi \operatorname{im}(s1)} \frac{\partial}{\partial s2}\operatorname{re}\left(e^{\frac{i\pi s2}{2}} e^{\frac{i\pi s3}{2}} e^{\frac{i\pi \overline{s2}}{2}} e^{\frac{i\pi \overline{s3}}{2}} \sin^2\left(\frac{\pi s4}{2}\right)\cos^2\left(\frac{\pi s4}{2}\right)\right)}{4} + \frac{\pi e^{\pi \operatorname{im}(s1)} e^{-\pi \operatorname{im}(s2)} e^{-\pi \operatorname{im}(s3)} \left|\cos^2\left(\frac{\pi s4}{2}\right)\right.}{4} \\
&-\frac{\pi e^{\pi \operatorname{im}(s2)} \operatorname{im}\left(e^{-\frac{i\pi s1}{2}} e^{-\frac{i\pi s3}{2}} e^{-\frac{i\pi \overline{s1}}{2}} e^{-\frac{i\pi \overline{s3}}{2}} \sin^2\left(\frac{\pi s4}{2}\right)\sin\left(\frac{\pi \overline{s4}}{2}\right)\cos\left(\frac{\pi \overline{s4}}{2}\right)\right)\frac{d}{ds2}\operatorname{im}(s_2)}{4} \\
&+\frac{\pi e^{\pi \operatorname{im}(s2)} \operatorname{im}\left(e^{\frac{i\pi s1}{2}} e^{\frac{i\pi s3}{2}} e^{\frac{i\pi \overline{s1}}{2}} e^{\frac{i\pi \overline{s3}}{2}} \sin\left(\frac{\pi s4}{2}\right)\sin^2\left(\frac{\pi \overline{s4}}{2}\right)\cos\left(\frac{\pi \overline{s4}}{2}\right)\right)\frac{d}{ds2}\operatorname{im}(s_2)}{4} \\
&-\frac{e^{\pi \operatorname{im}(s3)} \frac{\partial}{\partial s2}\operatorname{im}\left(e^{-\frac{i\pi s1}{2}} e^{-\frac{i\pi s2}{2}} e^{-\frac{i\pi \overline{s1}}{2}} e^{-\frac{i\pi \overline{s2}}{2}} \sin^2\left(\frac{\pi s4}{2}\right)\sin\left(\frac{\pi \overline{s4}}{2}\right)\cos\left(\frac{\pi \overline{s4}}{2}\right)\right)}{4} \\
&+\frac{e^{\pi \operatorname{im}(s3)} \frac{\partial}{\partial s2}\operatorname{im}\left(e^{\frac{i\pi s1}{2}} e^{\frac{i\pi s2}{2}} e^{\frac{i\pi \overline{s1}}{2}} e^{\frac{i\pi \overline{s2}}{2}} \sin\left(\frac{\pi s4}{2}\right)\sin^2\left(\frac{\pi \overline{s4}}{2}\right)\cos\left(\frac{\pi \overline{s4}}{2}\right)\right)}{4} \\
&+\frac{e^{-\pi \operatorname{im}(s3)} \frac{\partial}{\partial s2}\operatorname{im}\left(e^{-\frac{i\pi s1}{2}} e^{\frac{i\pi s2}{2}} e^{-\frac{i\pi \overline{s1}}{2}} e^{\frac{i\pi \overline{s2}}{2}} \sin\left(\frac{\pi \overline{s4}}{2}\right)\cos^2\left(\frac{\pi s4}{2}\right)\cos\left(\frac{\pi \overline{s4}}{2}\right)\right)}{4} \\
&-\frac{e^{-\pi \operatorname{im}(s3)} \frac{\partial}{\partial s2}\operatorname{im}\left(e^{\frac{i\pi s1}{2}} e^{-\frac{i\pi s2}{2}} e^{\frac{i\pi \overline{s1}}{2}} e^{-\frac{i\pi \overline{s2}}{2}} \sin\left(\frac{\pi s4}{2}\right)\cos\left(\frac{\pi s4}{2}\right)\cos^2\left(\frac{\pi \overline{s4}}{2}\right)\right)}{4} \\
&-\frac{\pi e^{-\pi \operatorname{im}(s2)} \operatorname{im}\left(e^{-\frac{i\pi s1}{2}} e^{\frac{i\pi s3}{2}} e^{-\frac{i\pi \overline{s1}}{2}} e^{\frac{i\pi \overline{s3}}{2}} \sin\left(\frac{\pi \overline{s4}}{2}\right)\cos^2\left(\frac{\pi s4}{2}\right)\cos\left(\frac{\pi \overline{s4}}{2}\right)\right)\frac{d}{ds2}\operatorname{im}(s_2)}{4}
\end{aligned}
$$

But this circuit was toooooo complicated to implement! (cause we needed 138 qubits…)

So we used only this part (for now!)



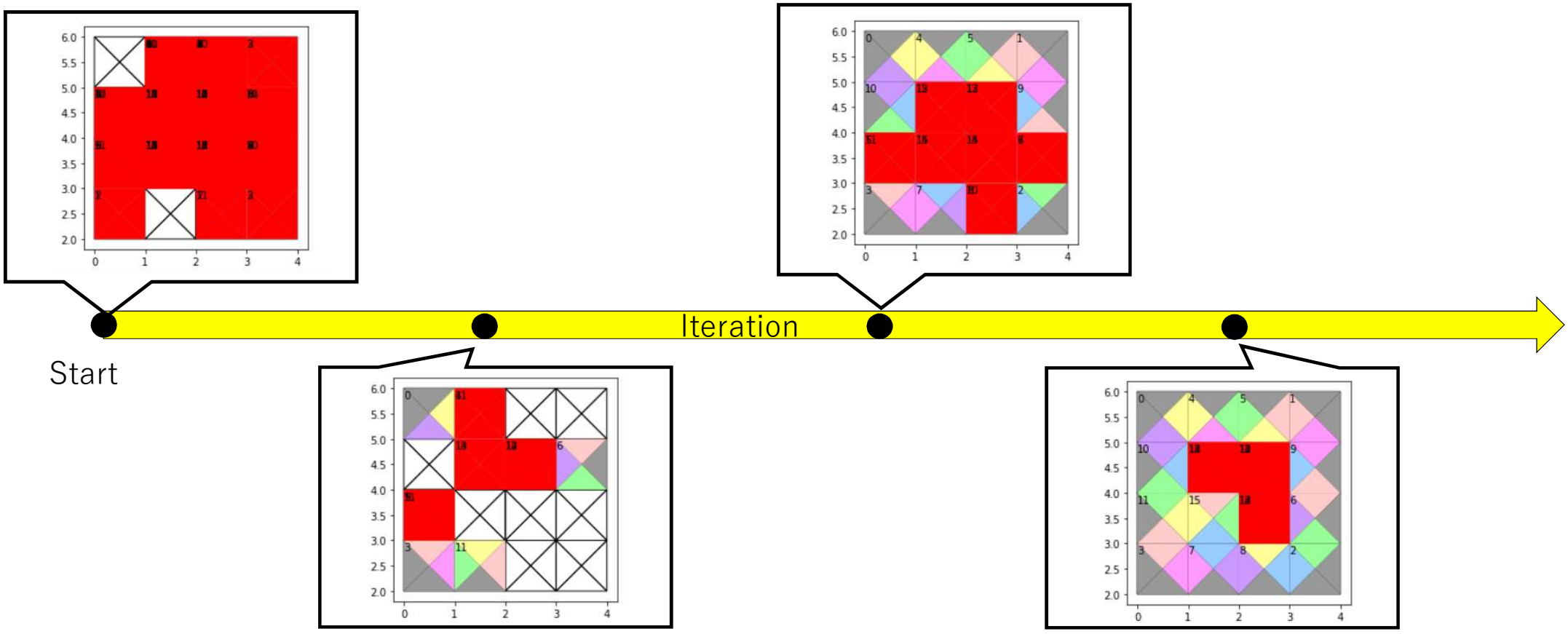
We gave up using connection between 2 qubits for now

Implementation

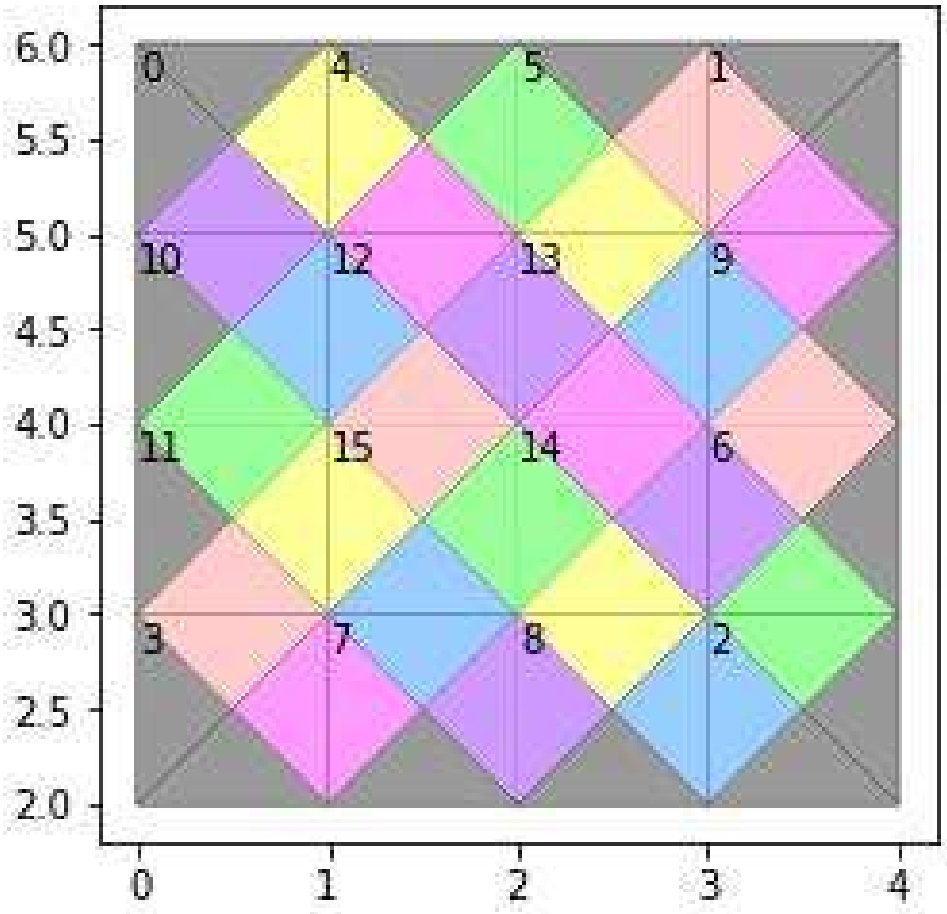
We implemented the circuit,
Evaluation function (with traditional computer)
And whole process

and run the program

The progress of the solving process…



Finally we solved the puzzle! (4x4 though)

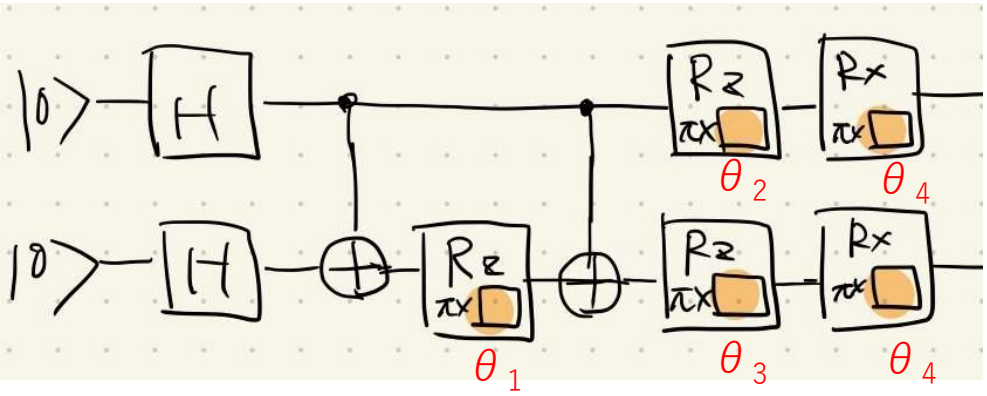


Iteration: 131

result: [1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]

To improving the result

Implementing the circuits for connection between two qubits



Start from complete random θ_i

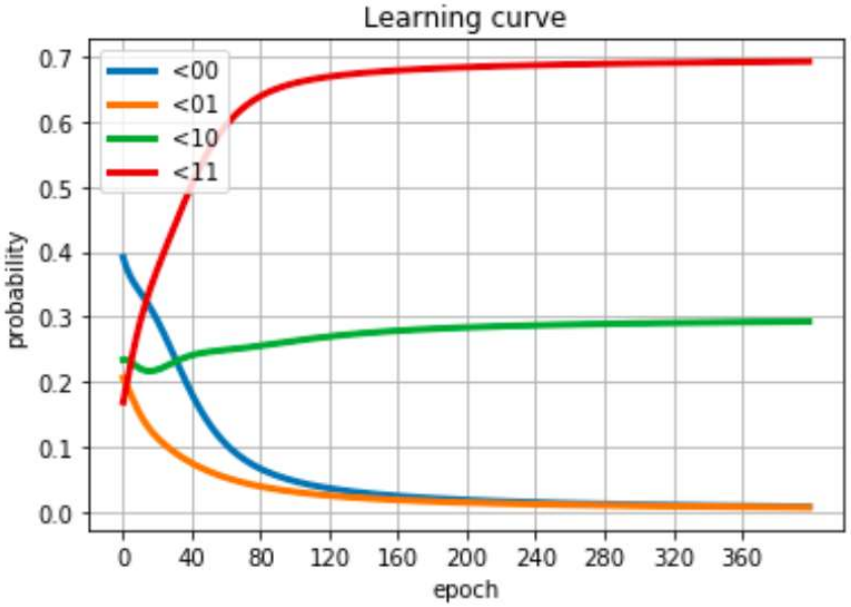
(1) Generate the sample from the circuit

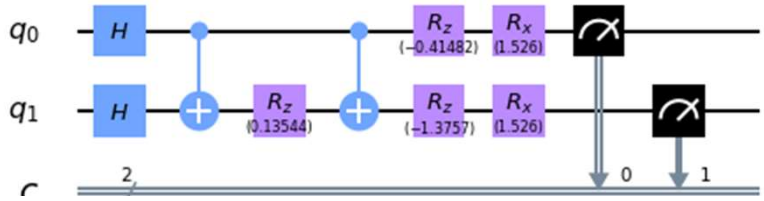
(2) Calculate the gradient by
partial differentiation for each rotation θ_i

(3) Adjust the rotation θ_i with the gradient

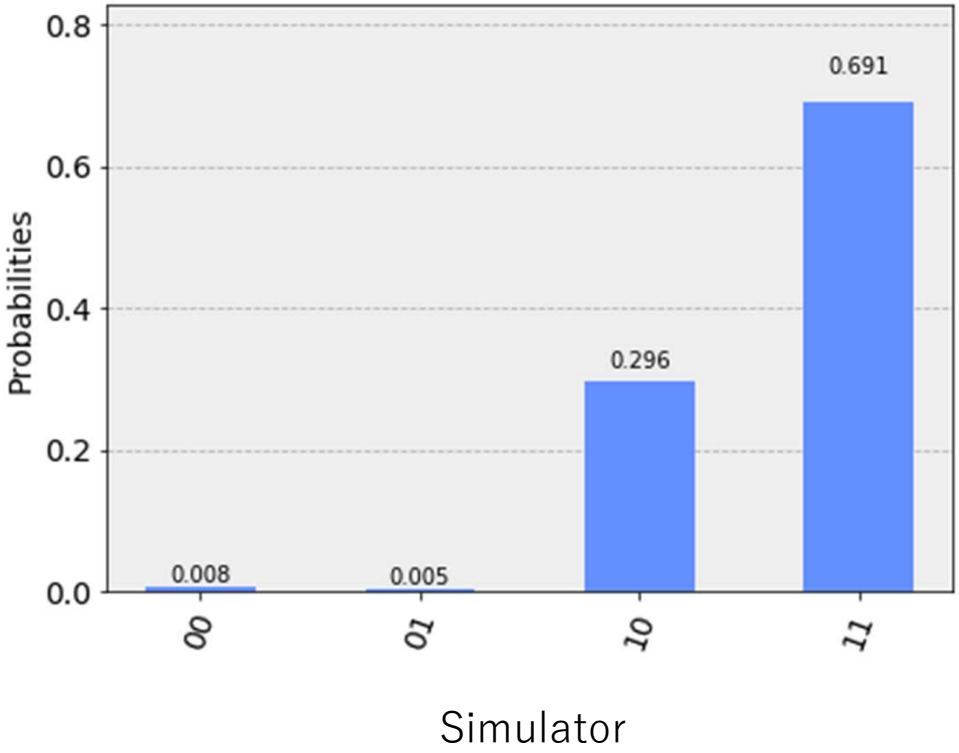
Then iterate from (1) to (3)

We aimed to adjust the circuit to generate the sample
|11 > 70%, |10> 30%
And we could adjust the circuit like this!

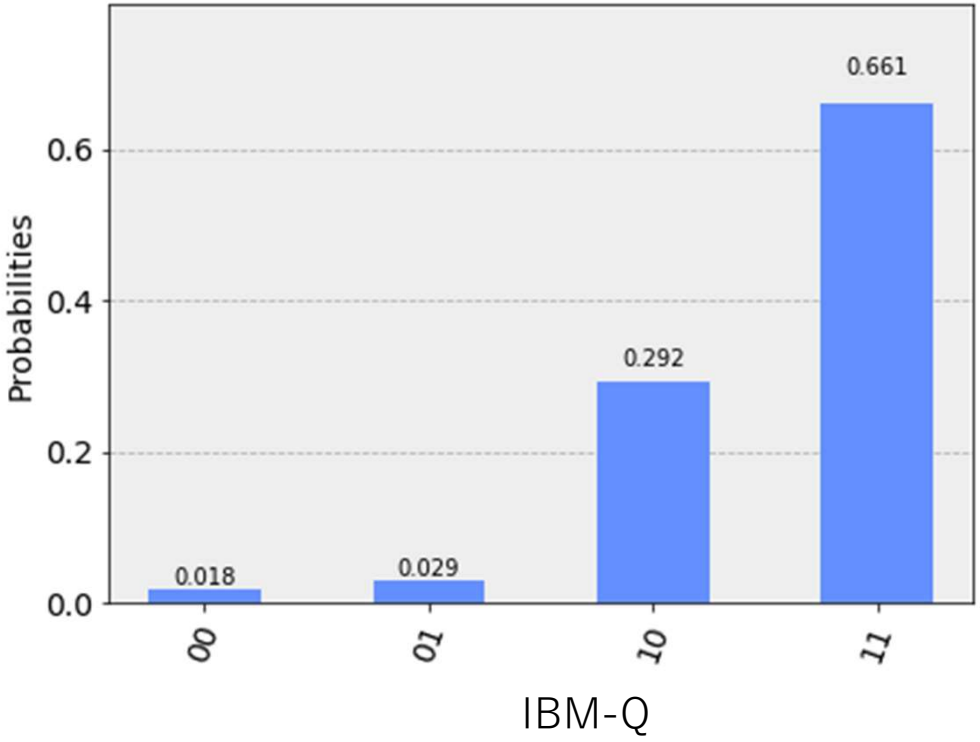




Sampling from the adjusted circuit



Simulator



IBM-Q

Future tasks

- Implementing full circuit using connection of two qubits
- Solving larger puzzles (next 5x5)
- Try other aproaches

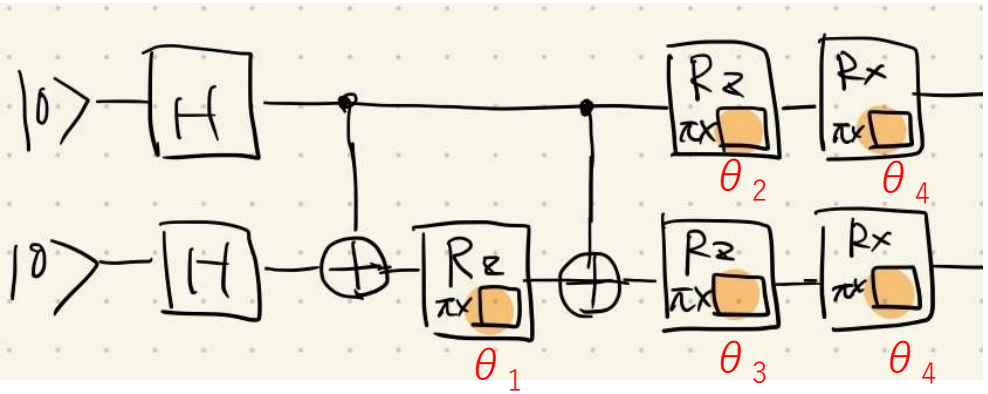
Finally we will solve whole Eternity II
(In the future!)

Thank you!

Akiyama
Yuki Naito
Azegami
And Asa Eagle

Additional work

Implementing the circuits for connection between two qubits



Start from complete random θ_i

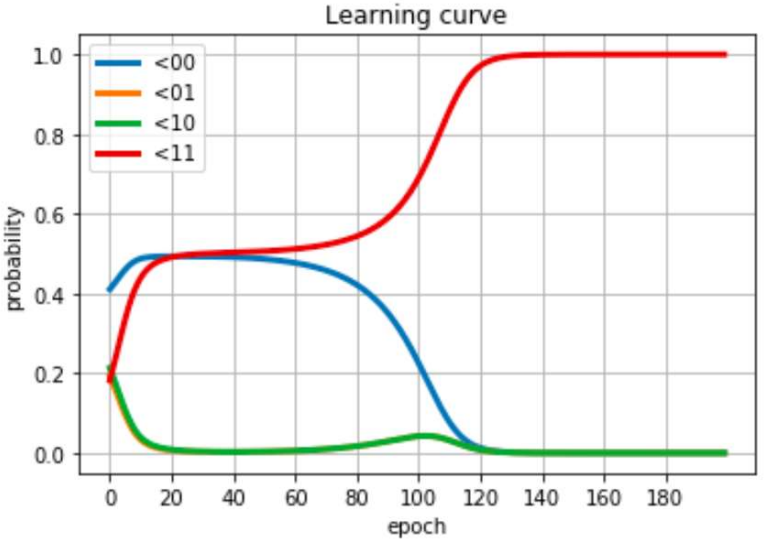
(1) Generate the sample from the circuit

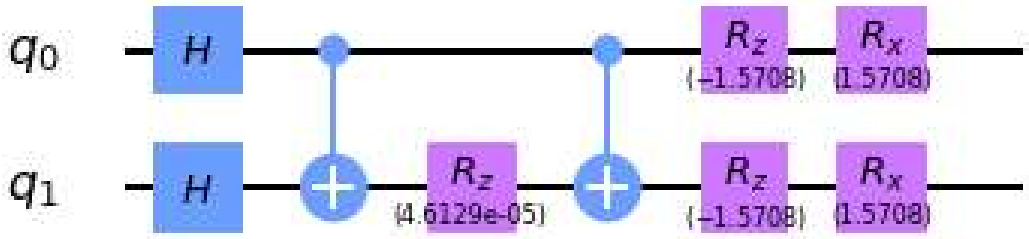
(2) Calculate the gradient by
partial differentiation for each rotation θ_i

(3) Adjust the rotation θ_i with the gradient

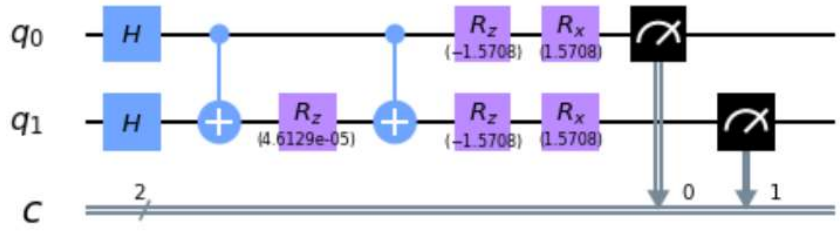
Then iterate from (1) to (3)

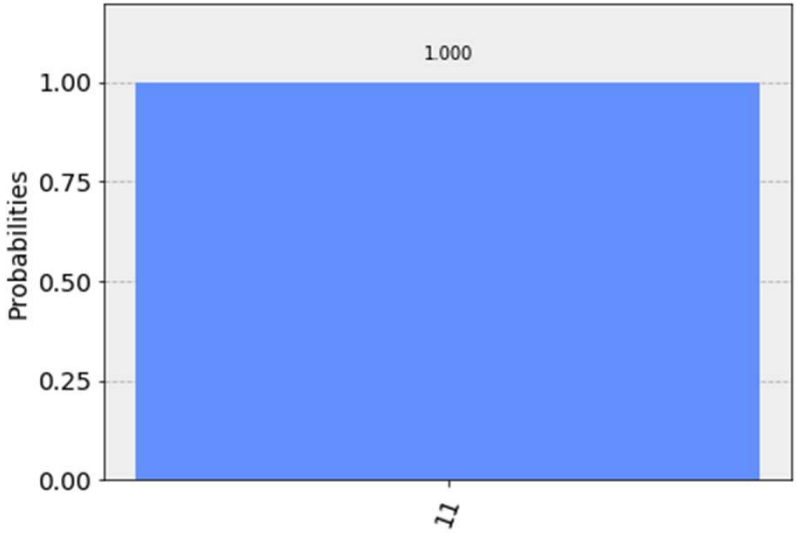
We aimed to adjust the circuit to generate the sample |11>
And we could adjust the circuit like this!



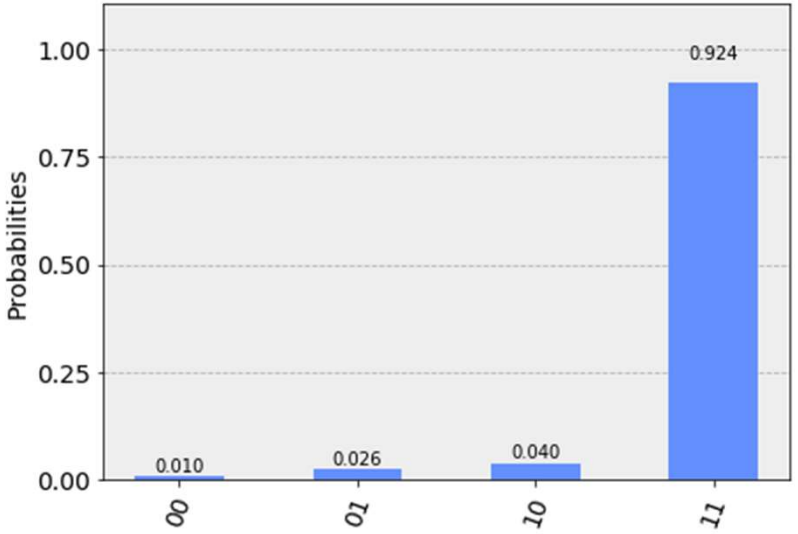


Sampling from the adjusted circuit





Simulator



IBM-Q