Quantum Reinforcement Learning

Group 12

吳冠緯、吳育丞、陳宥辰

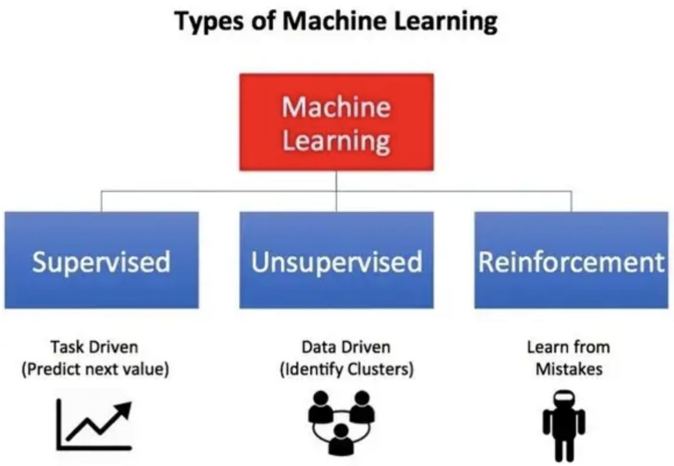
Outline

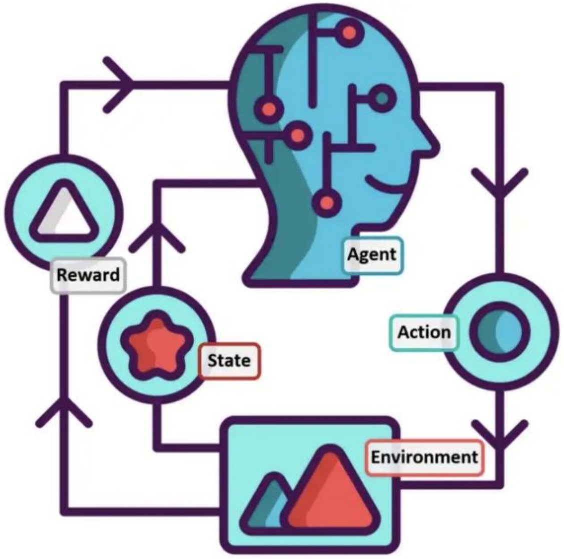
- Value-based Reinforcement Learning
- Quantum Reinforcement Learning
- Another Quantum Application with RL

Value-based Reinforcement Learning

What's Reinforcement Learning?

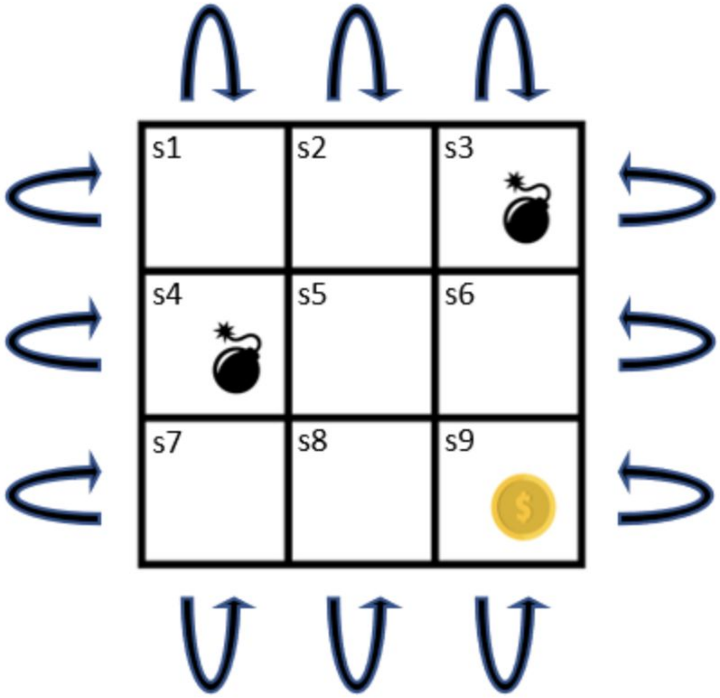
- **Reinforcement Learning** is a method of machine learning by which an algorithm can make decisions and **take actions within a given environment**, and **learns what appropriate decisions to make through repeated trial-and-error actions**.





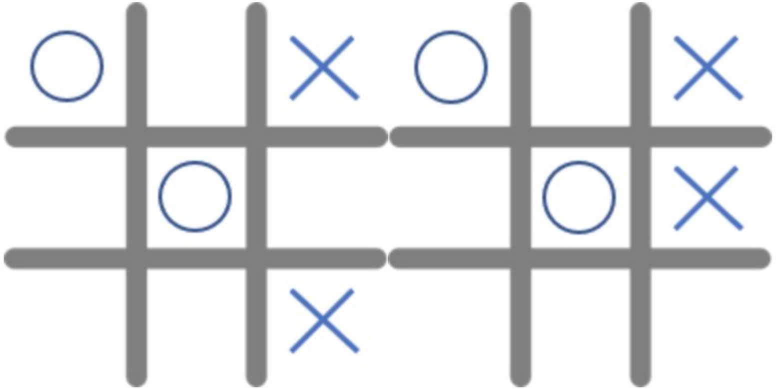
Example of Reinforcement Learning

- State:
  - S1, S2, ..., S9
- Action:
  - Move up
  - Move down
  - Move right
  - Move left
- Reward:
  - Money: 1
  - Bomb: -1
  - Otherwise: 0



Markov Decision Process

- A **discrete-time** stochastic control process.
- A framework for modeling decision making in situations where **the outcomes are partly random and partly under control of the decision maker**.
- **The state transitions of an MDP are independent of all previous states.**



Modeling RL with MDP

- State (S)
- Action (A)
- Reward (R)
- Policy (pi)

$$\pi(a \mid s)$$

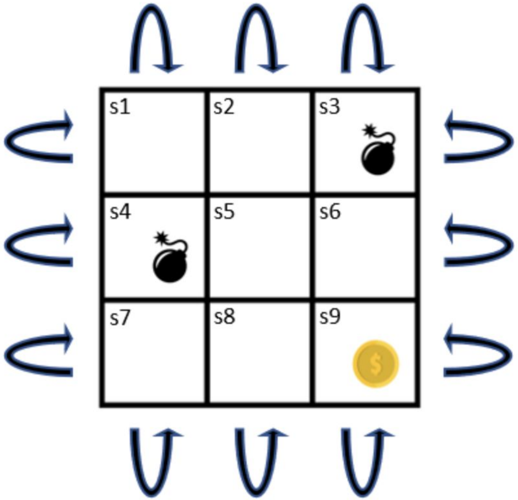
- Objective (G)

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots + \gamma^{T-t-1} R_T$$

Value Function

- Formulate the objective more precisely:

$$
\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\
&= \mathbb{E}_\pi\left[R_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k R_{t+k+2} \mid S_t = s\right] \\
&= \mathbb{E}_\pi\left[R_{t+1} + \gamma V_\pi(s') \mid S_{t+1} = s'\right]
\end{aligned}
$$



- Example:

$$V_\pi(S_6) = \frac{1}{4}(-1 + 0.7 * V_\pi(S_3)) + \frac{1}{4}(+0 + 0.7 * V_\pi(S_5)) + \frac{1}{4}(+0 + 0.7 * V_\pi(S_6)) + \frac{1}{4}(+1 + 0.7 * V_\pi(S_9))$$

How to get optimal policy?

- Greedy Action

$$V_\pi(s) = \sum_{a \in A(s)} q_\pi(s,a) \Rightarrow \pi'(s) \doteq \arg\max_a q_\pi(s,a)$$

$$V_\pi(S_6) = \frac{1}{4}(-1+0.7*V_\pi(S_3)) + \frac{1}{4}(+0+0.7*V_\pi(S_5)) + \frac{1}{4}(+0+0.7*V_\pi(S_6)) + \frac{1}{4}(+1+0.7*V_\pi(S_9))$$

- Policy Iteration

$$\pi_0 \xrightarrow{\mathrm{E}} v_{\pi_0} \xrightarrow{\mathrm{I}} \pi_1 \xrightarrow{\mathrm{E}} v_{\pi_1} \xrightarrow{\mathrm{I}} \pi_2 \xrightarrow{\mathrm{E}} \cdots \xrightarrow{\mathrm{I}} \pi_* \xrightarrow{\mathrm{E}} v_*,$$

- $\xrightarrow{\mathrm{E}}$ 稱為policy evaluation
- $\xrightarrow{\mathrm{I}}$ 稱為policy improvement。

→ How to accelerate the process ?

Dynamic Programming

Input π, the policy to be evaluated
Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
    $\Delta \leftarrow 0$
    Loop for each $s \in \mathcal{S}$:
        $v \leftarrow V(s)$
        $V(s) \leftarrow \boxed{\sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\left[r + \gamma V(s')\right]}$
        $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

→ Can we always know the value function in advance ?

Monte Carlo Method and Temporal Difference Learning

$$
\begin{aligned}
V_{n+1}(s) &= \mathbb{E}_\pi[G_t \mid S_t = s] \\
&= \frac{G_{t1} + G_{t2} + G_{t3} + \ldots + G_{tn}}{n} \\
&= \frac{1}{n}(G_{tn} + (n-1)\frac{1}{n-1}\sum_{i=1}^{n-1} G_{ti}) \\
&= \frac{1}{n}(G_{tn} + (n-1)V(s)) \\
&= \frac{1}{n}(G_{tn} + nV(s) - V(s)) \\
&= V_n(s) + \frac{1}{n}[G_{tn} - V_n(s)]
\end{aligned}
$$

$$V(S_t) \leftarrow V(S_t) + \alpha\Big[G_t - V(S_t)\Big]$$

$$\boxed{V(S_t) \leftarrow V(S_t) + \alpha\Big[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\Big]}$$

Final formulation of updating value function

Quantum Reinforcement Learning

Define Actions with Qubit States

- An action set can be represented by a superposition state of n qubits.

$$N_a \leq 2^n \leq 2N_a \quad |A\rangle = \sum_n \beta_n |a_n\rangle \quad \sum_n |\beta_n|^2 = 1$$

- When we measure the qubit, it will collapse into one of its eigen action $|a_n\rangle$ with the probability of β_n .

Exploitation vs Exploration in RL

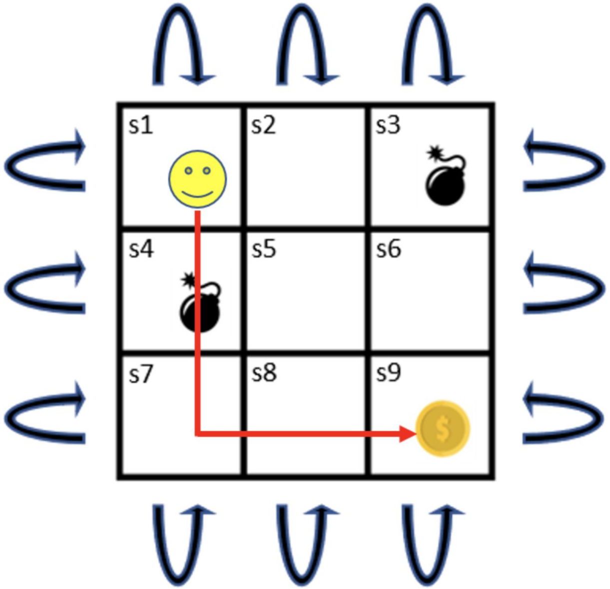
- Greedy action (exploitation)

$$\pi'(s) \doteq \arg\max_a q_\pi(s,a)$$

- Epsilon-greedy (exploration)

  A probability of the agent randomly conducting an action rather than greedy action.

  → But how?



Utilize Grover's Algorithm

- Start with equally weighted superposition (n Hadamard gates)

$$|a_0^{(n)}\rangle = \frac{1}{\sqrt{2^n}}\left(\sum_{a=00\cdots0}^{\overbrace{11\cdots1}^{n}} |a\rangle\right)$$

- Amplitude amplification

$$\pi'(s) \doteq \underset{a}{\arg\max}\, q_\pi(s,a)$$

⇨ $U_a = I - 2|a\rangle\langle a|$

⇨ $U_{a_0^{(n)}} = H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n} = 2|a_0^{(n)}\rangle\langle a_0^{(n)}| - I$

QRL Algorithm

Procedural QRL:

Initialize $|s^{(m)}\rangle = \sum_{s=00\cdots0}^{11\cdots1} C_s |s\rangle$, $f(s) = |a_s^{(n)}\rangle = \sum_{a=00\cdots0}^{11\cdots1} C_a |a\rangle$ and $V(s)$ arbitrarily

Repeat (for each episode)

For all states $|s\rangle$ in $|s^{(m)}\rangle = \sum_{s=00\cdots0}^{11\cdots1} C_s |s\rangle$:

1. Observe $f(s) = |a_s^{(n)}\rangle$ and get $|a\rangle$;
2. Take action $|a\rangle$, observe next state $|s'\rangle$, reward r, then
   (a) Update state value: $V(s) \leftarrow V(s) + \alpha(r + \gamma V(s') - V(s))$
   (b) Update probability amplitudes:
   repeat U_{Grov} for L times

$$U_{Grov} |a_s^{(n)}\rangle = U_{a_0^{(n)}} U_a |a_s^{(n)}\rangle$$

Until for all states $|\Delta V(s)| \le \varepsilon$.

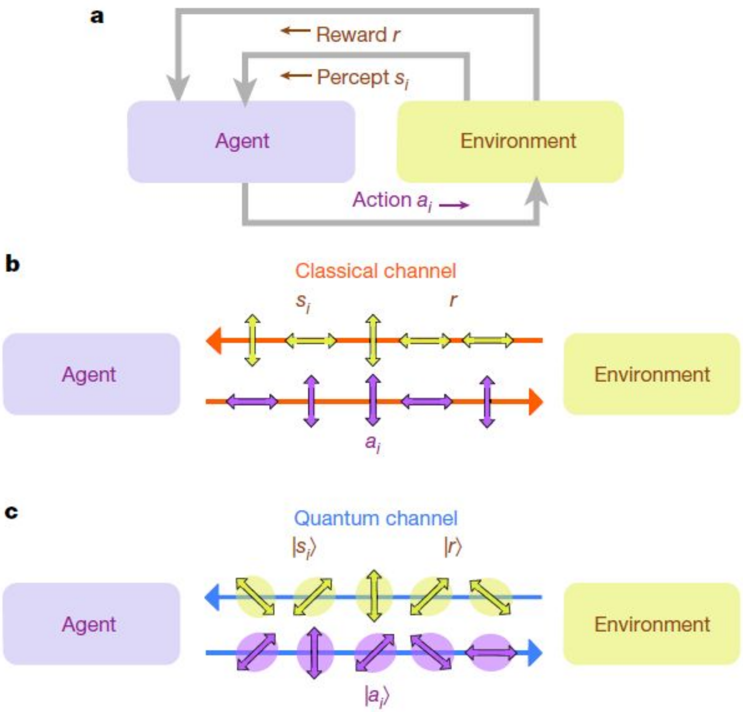
[1]

Another Quantum Application with RL

Schematic of Learning Agent

- Agent and environment interacting **classically**, where communication is only possible via a fixed preferred basis.

- Agent and environment interacting via a **quantum** channel, where arbitrary **superposition states** are exchanged.



[2]

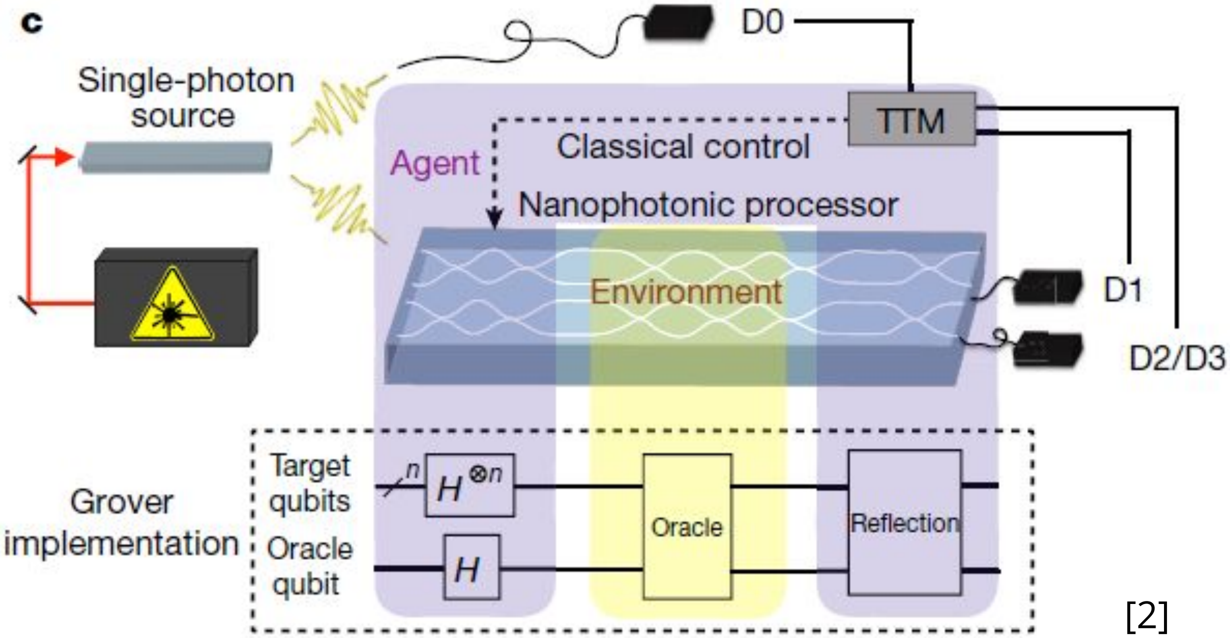
Framework

- Focus on so-called **deterministic strictly epochal** (DSE) learning scenarios.

- Here 'epochs' consist of strings of percepts **s = (s0, ..., sL−1)** with fixed s0, actions **a = (a1, ..., aL)** of fixed length L, and a final reward **r**, and both **s** = **s**(**a**) and **r** = **r**(**a**) are completely determined by **a**.

$$U_E|\mathbf{a}\rangle_A|0\rangle_R = \begin{cases} |\mathbf{a}\rangle_A|1\rangle_R \text{ if } r(\mathbf{a}) > 0 \\ |\mathbf{a}\rangle_A|0\rangle_R \text{ if } r(\mathbf{a}) = 0 \end{cases}.$$

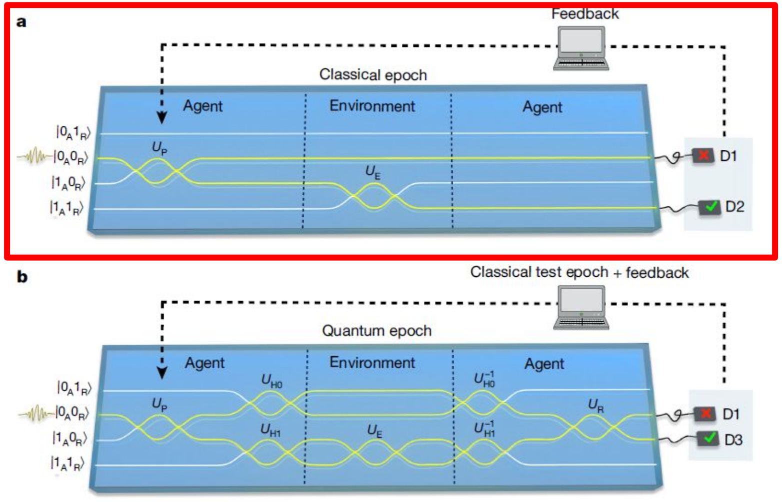
Experiment Setup



[2]

Classical Epoch

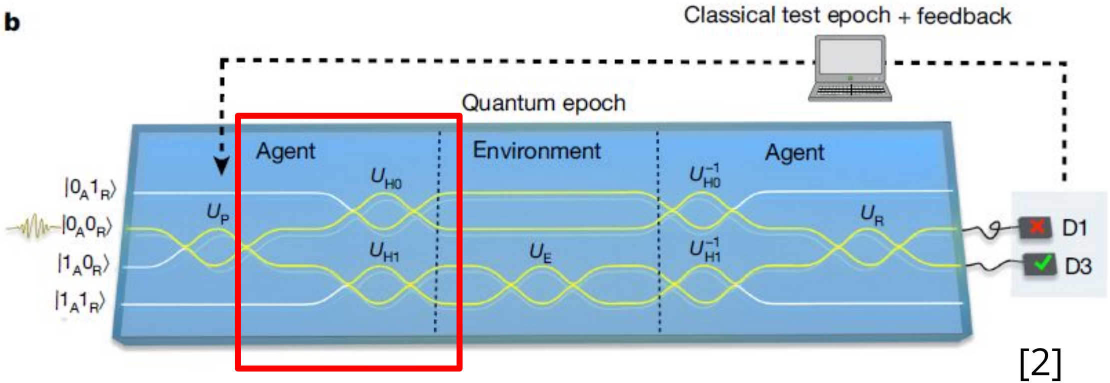
- In a classical strategy, the environment **flips** the reward qubit only if the action qubit is in the winning state via UE. Next, the photon is **coupled out and detected** in either D1 or D2 with probability cos2(ξ) and sin2(ξ), respectively.



[2]

Quantum Epoch

(1) The agent prepares the state $|\psi\rangle_A|-\rangle_R$, with $|\psi\rangle_A = \sum_a \sqrt{p(\mathbf{a})}\,|\mathbf{a}\rangle_A = \cos(\xi)|\ell\rangle_A + \sin(\xi)|w\rangle_A$, and sends it to the environment. $|w\rangle_A$ and $|\ell\rangle_A$ are superpositions of all winning (rewarded) and losing (non-rewarded) action sequences, respectively, and $|-\rangle_R = (|0\rangle_R - |1\rangle_R)/\sqrt{2}$.



[2]

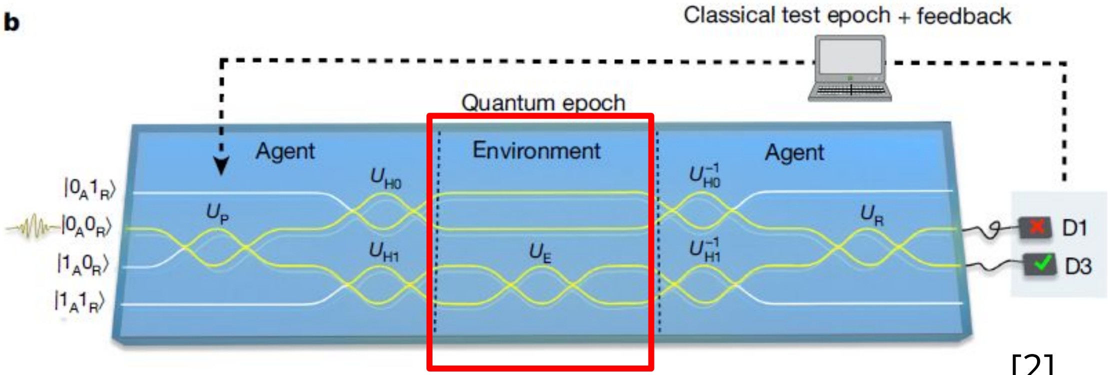
Quantum Epoch

(2) The environment applies U_E from equation (1) to $|\psi\rangle_A|-\rangle_R$, flipping the sign of the winning state:

$$U_E|\psi\rangle_A|-\rangle_R = [\cos(\xi)|\ell\rangle_A - \sin(\xi)|w\rangle_A]|-\rangle_R, \quad (3)$$

and returns the resulting state to the agent.

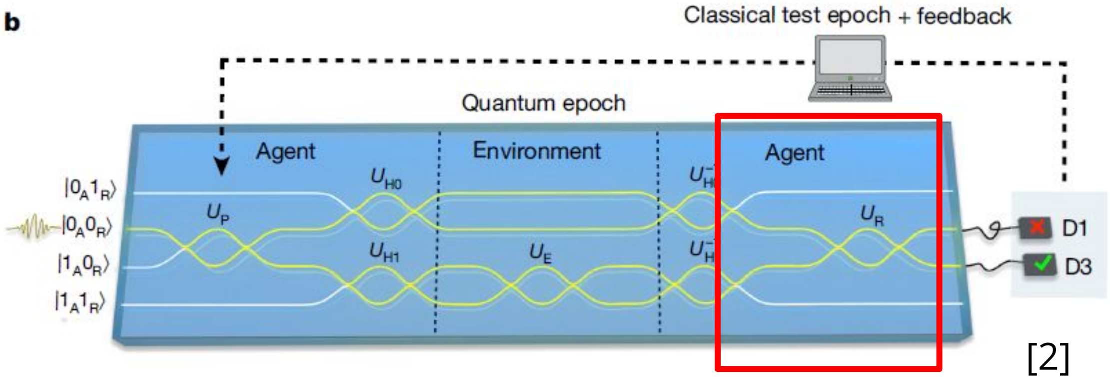
b



[2]

Quantum Epoch

(3) The agent performs a reflection $U_R = 2|\psi\rangle\langle\psi|_A - \mathbb{1}_A$ over the initial state $|\psi\rangle_A$.

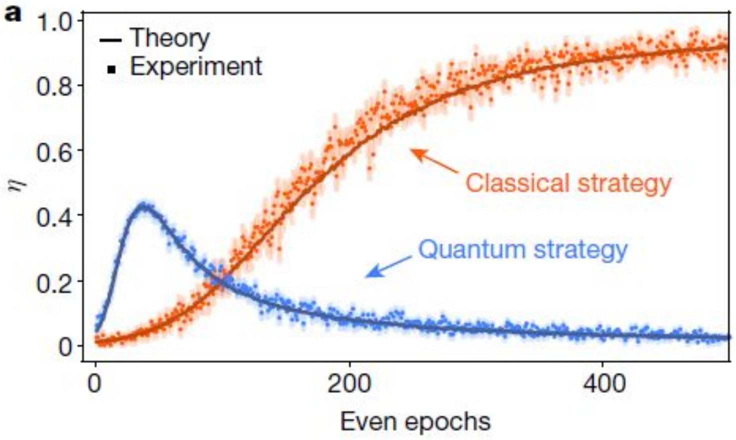


[2]

Improvement in Learning Time

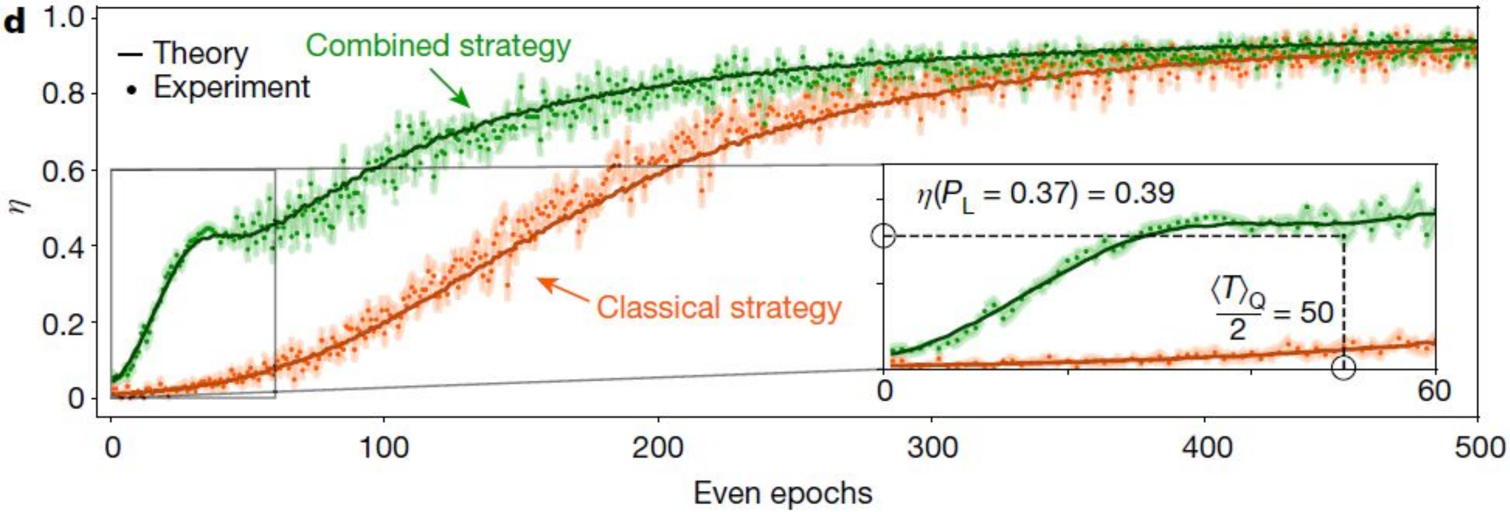
- For ε < P, **ηQ > ηC**, meaning that the quantum strategy proves advantageous over the classical case.

- However, as soon as **ηQ = ηC** (at P = 0.396), a classical agent starts outperforming a quantum-enhanced agent that still performs Grover iterations.



[2]

Combined Strategy



Topics Candidates

- Implement QRL algorithm in work 1 on different RL tasks.
- Combine the advantages of previous two works on certain RL tasks.
- Implement QNN for RL tasks.

Reference

[1] Dong, D.; Chen, C.; Li, H.; Tarn, T.-J. Quantum Reinforcement Learning. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) 2008, 38, 1207.

[2] Saggio, V., Asenbeck, B.E., Hamann, A. *et al.* Experimental quantum speed-up in reinforcement learning agents. *Nature* 591, 229–233 (2021).