

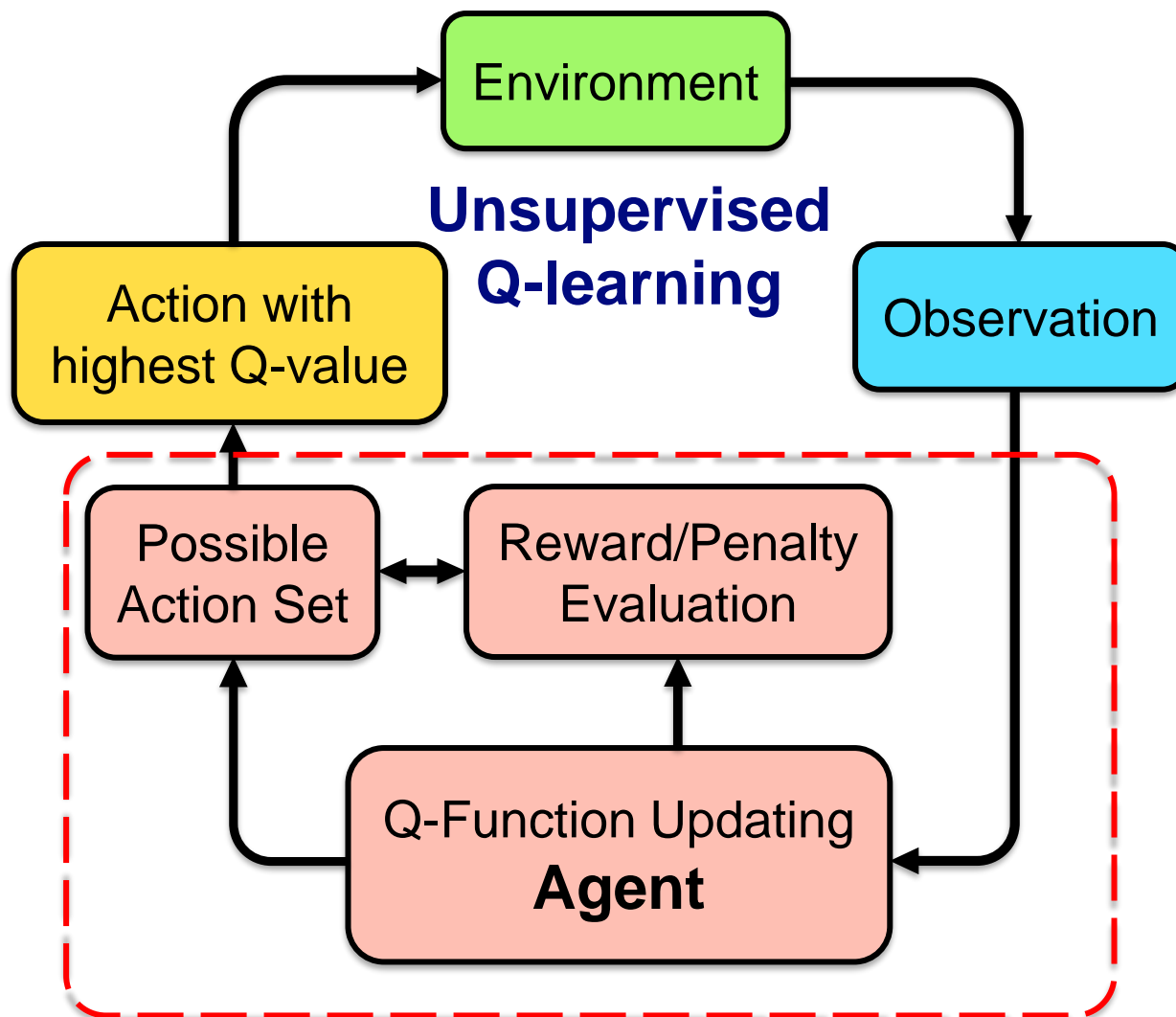
Grover Method for Quantum Reinforcement Learning

Zhanzhi Jiang and Qiang Gao

Outline

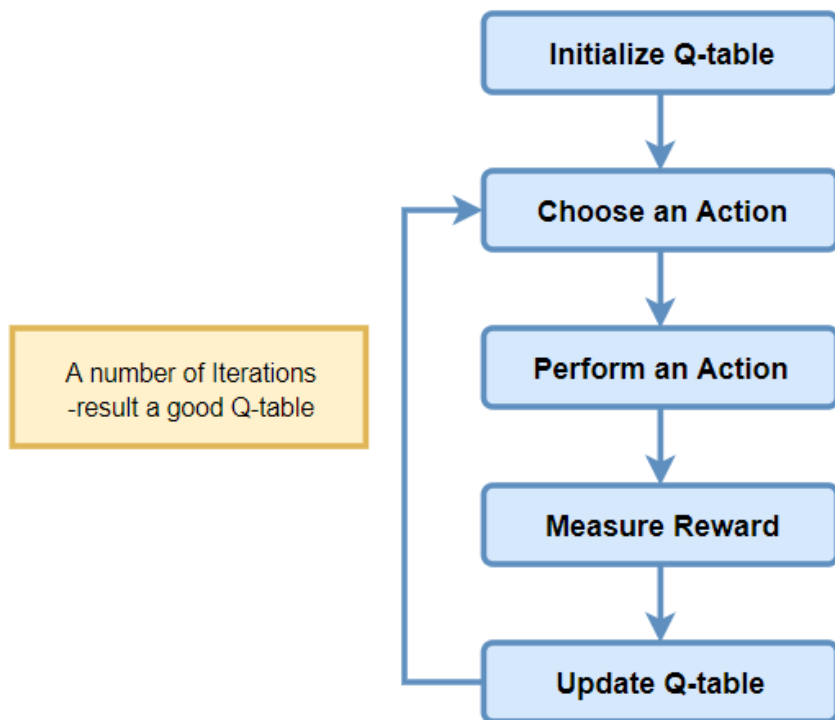
- ❖ Introduction to reinforcement learning (RL)
- ❖ Quantum RL based on Grover algorithm
- ❖ Test with simple environments
- ❖ Summary and future directions

Reinforcement learning



Q-learning

$Q(s, a)$: expectation of total reward for taking action a at state s



Update Q-table:

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha \left(R(s, a) + \gamma \max_{a'} Q(s', a') \right)$$

α : learning rate

$R(s, a)$: reward for taking action a at state s

γ : discount factor for convergence

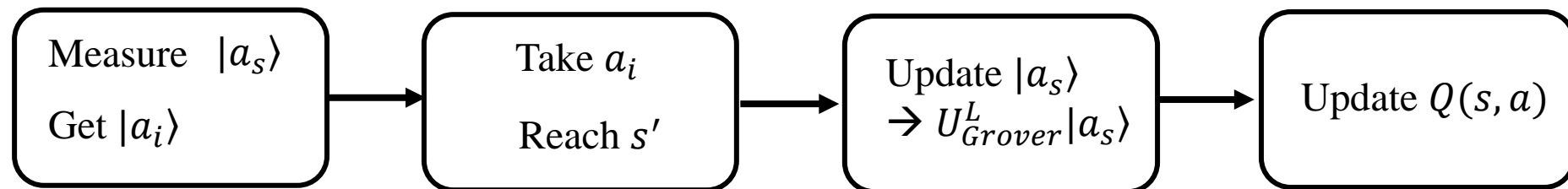
Method to implement Q-learning agents:

- Direct searching the Q-table
- Decision Tree and Deep Neural Networks (Alpha Go)
- Quantum agent: Grover amplitude amplification

Quantum Q-Learning Agent

Actions of every state s are encoded in a quantum state $|a_s\rangle = \sum_i c_i |a_i\rangle$

Take an action: the Grover iteration



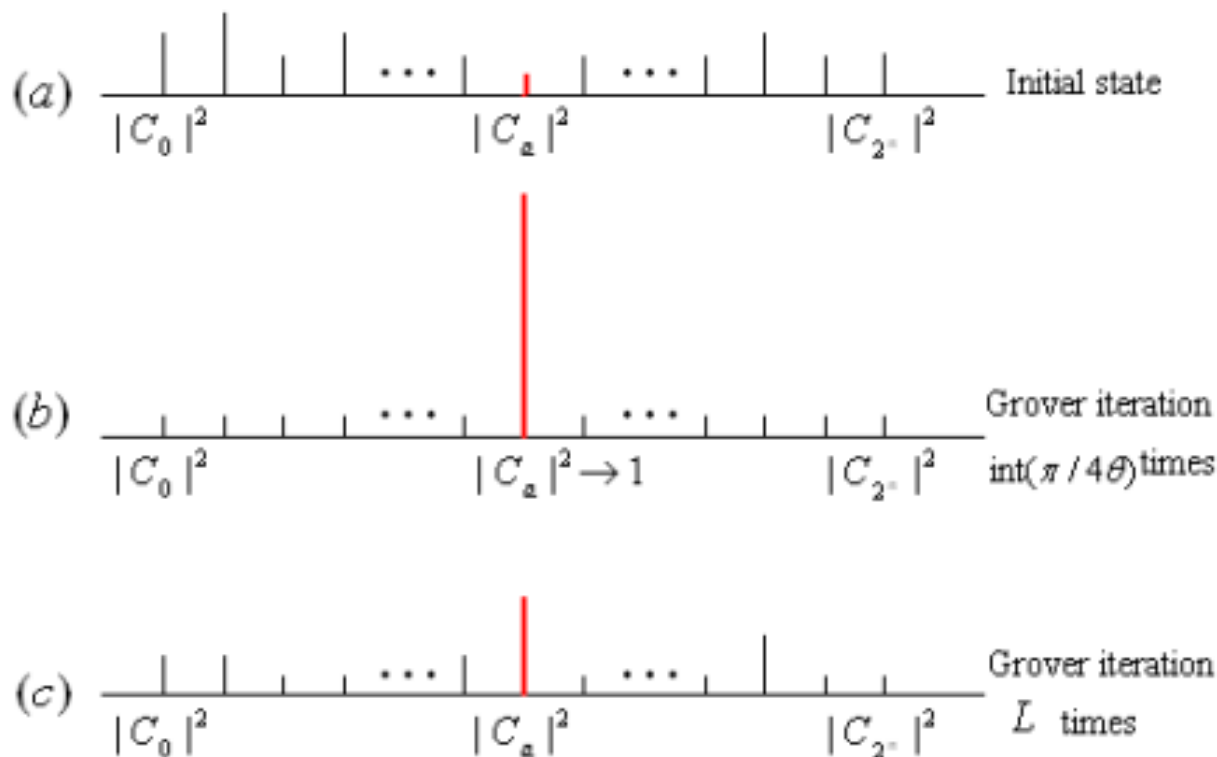
U_{Grover}^L : Grover oracle for performing the Grover quantum circuits

$$L = \text{int} \left[\min \left\{ k(r + V(s)), \frac{\pi}{4\theta} - \frac{1}{2} \right\} \right]: \text{Grover length}$$

a tuning parameter k (red circle)
 reward r (green circle)
 value function $V(s)$ (yellow box)
 a parameter related to the state space dimension θ (purple circle)

$$V(s) = \max_a Q(s, a)$$

How Grover algorithm works



- ❖ Grover Searching Algorithm: amplify the amplitude of target state to 1
- ❖ Grover Q-learning agent: amplify the amplitude of the action according to $Q(s', a)$

Quantum Q-learning algorithm

Algorithm 3 Quantum Q-learning (QQRL)

```
1: for all episodes do
2:   for all  $s \in S$  do
3:     Observe  $a$  from  $|a_s\rangle$ 
4:     Take action  $a$ , observe next state  $s'$  and reward  $r$ 
5:      $V(s') = \max_{a'} Q(s', a')$ 
6:      $L = \lfloor \min \{ k(r + V(s')), \frac{\pi}{4\theta} - \frac{1}{2} \} \rfloor$ 
7:      $|a_s\rangle \leftarrow \hat{U}_g^L |a_s\rangle$  ▷ Grover's Algorithm
8:      $Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma V(s') - Q(s, a))$ 
9:   end for
10: end for
```

Modifications:

The update of the V-values requires a search for maximal values in Q-table. To avoid that, we directly update $V(s)$ using $Q(s, a)$ with $|a\rangle$ being the measured results when given state s .

The quantum mechanics ensure us with high possibilities to get $\max_a Q(s, a)$.

Thus, we achieve a full-quantum Q-learning algorithm which might overperform its classical counterpart when the state space is large enough.

Frozen Lake



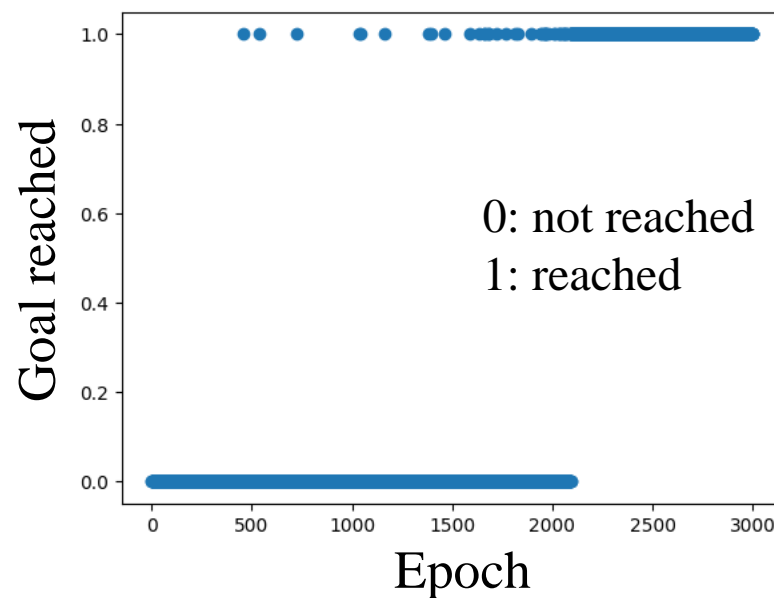
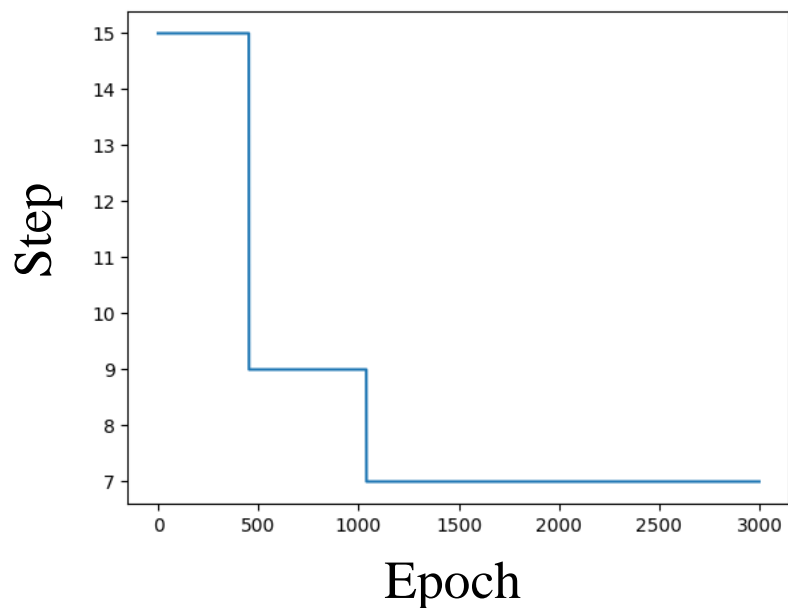
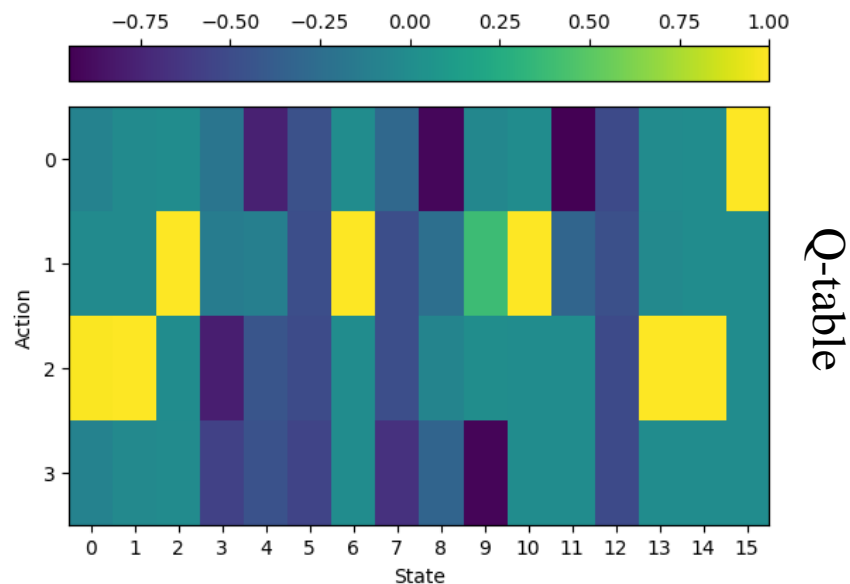
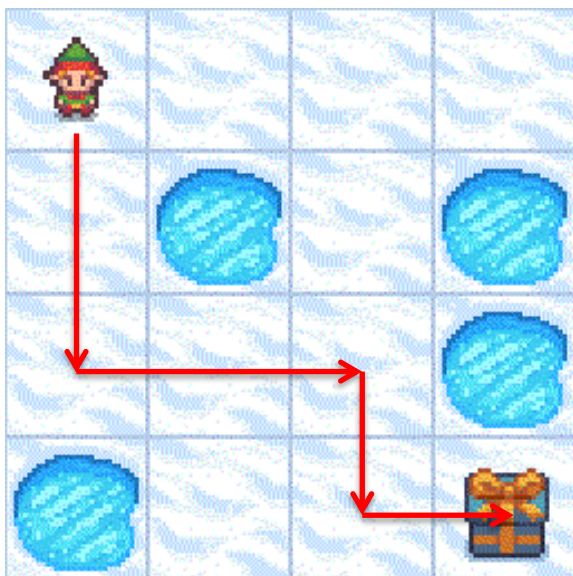
State: site location 0-15 (global env)

Action: move up, down, left, right

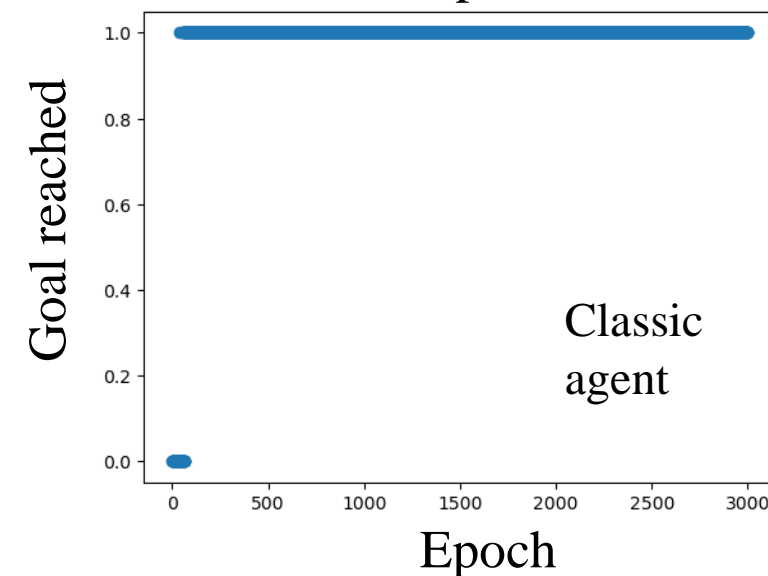
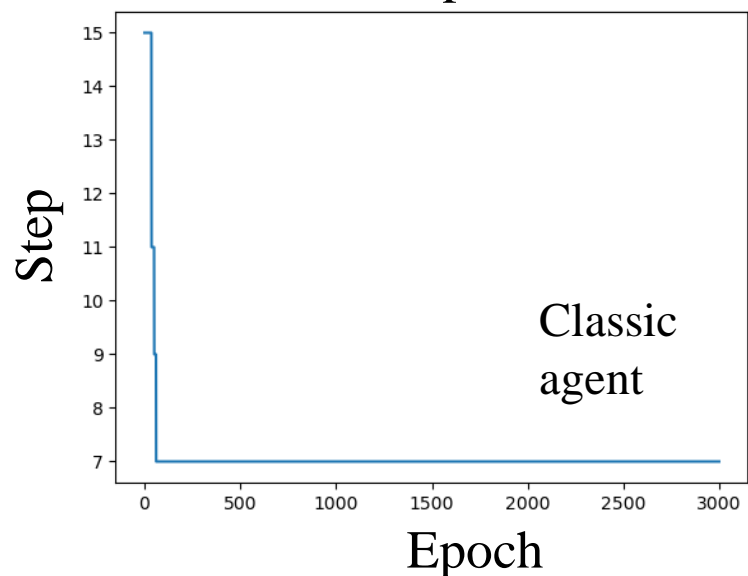
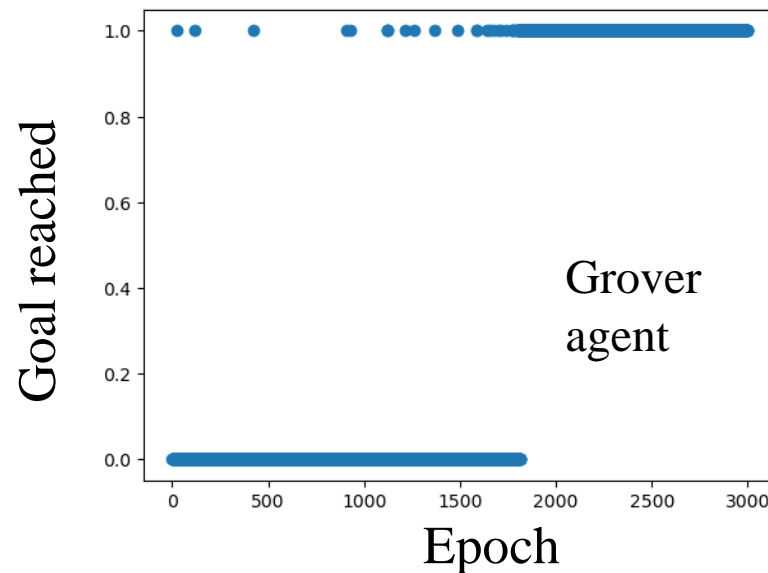
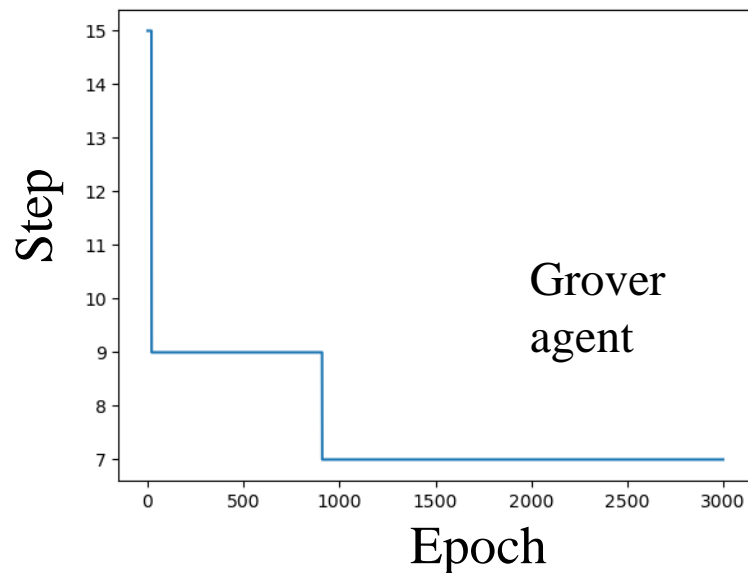
| Next State | Reward |
|------------|----------|
| Goal | 99, Done |
| Same site | -10 |
| Ice | -1 |
| Hole | 0, Done |

Task: find the optimal path from the start point (site 0) to the goal (site 15) without falling in the hole

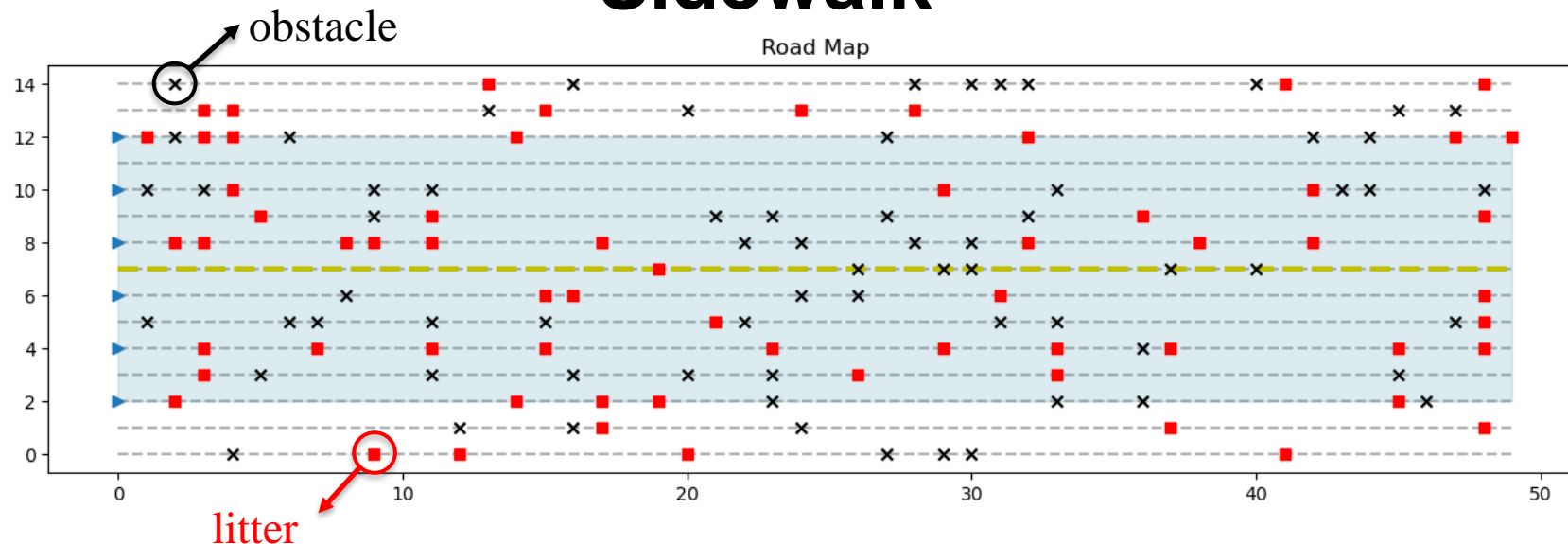
Result



Compare with classic agent



Sidewalk



Task: go through the sidewalk while picking up the litter and avoiding obstacles

State: four neighbors having litters (obstacles) or not, labeled by 0-15 (local env)

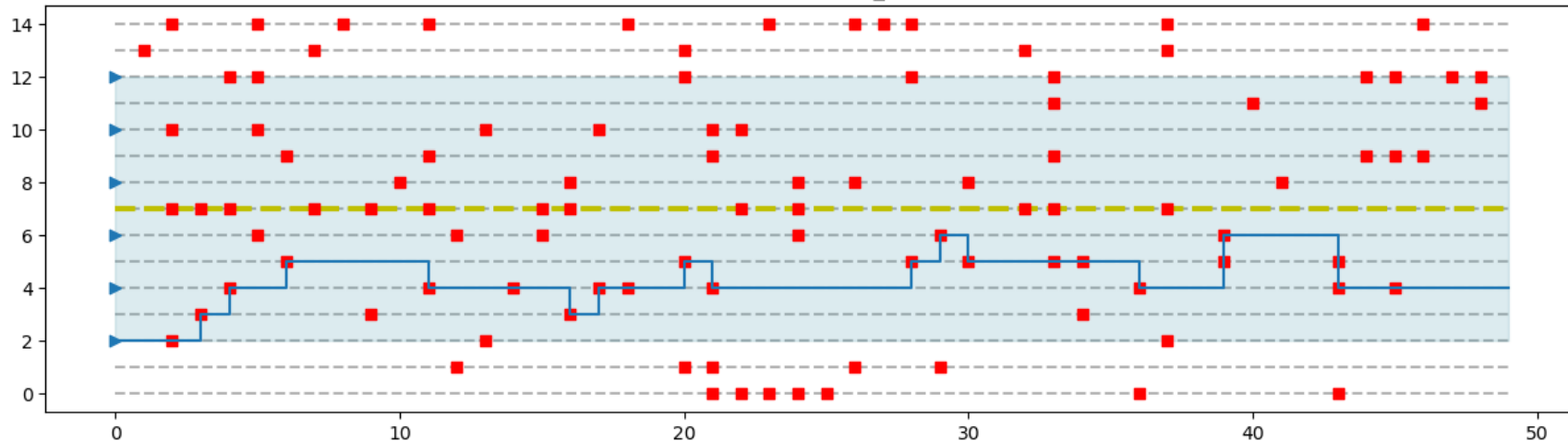
Action: move up, down, left, right

| Next position | Reward |
|--|--------|
| Litter (moving $\rightarrow, \uparrow, \downarrow$) | 15 |
| No litter (moving \rightarrow) | 5 |
| No litter (moving \uparrow, \downarrow) | 0 |
| Moving \leftarrow | -5 |

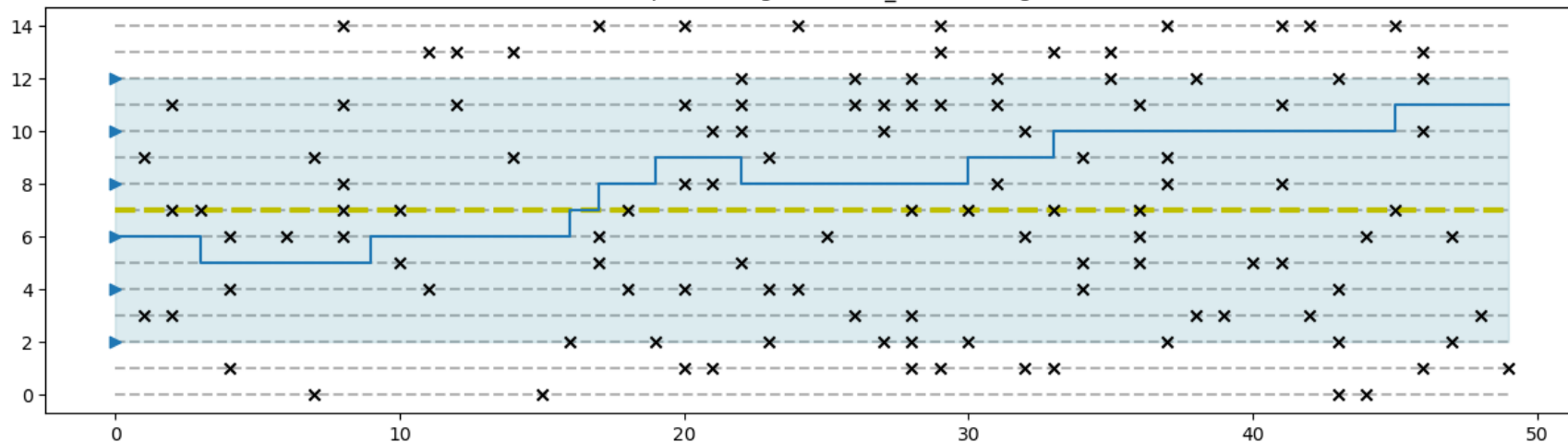
| Next position | Reward |
|--|--------|
| Obstacle | -3 |
| No obstacle (moving \rightarrow) | 8 |
| No obstacle (moving \uparrow, \downarrow) | 3 |
| No obstacle (moving \leftarrow) | 1 |

Quantum Agent

Road Map: picking up litters_Quantum agent

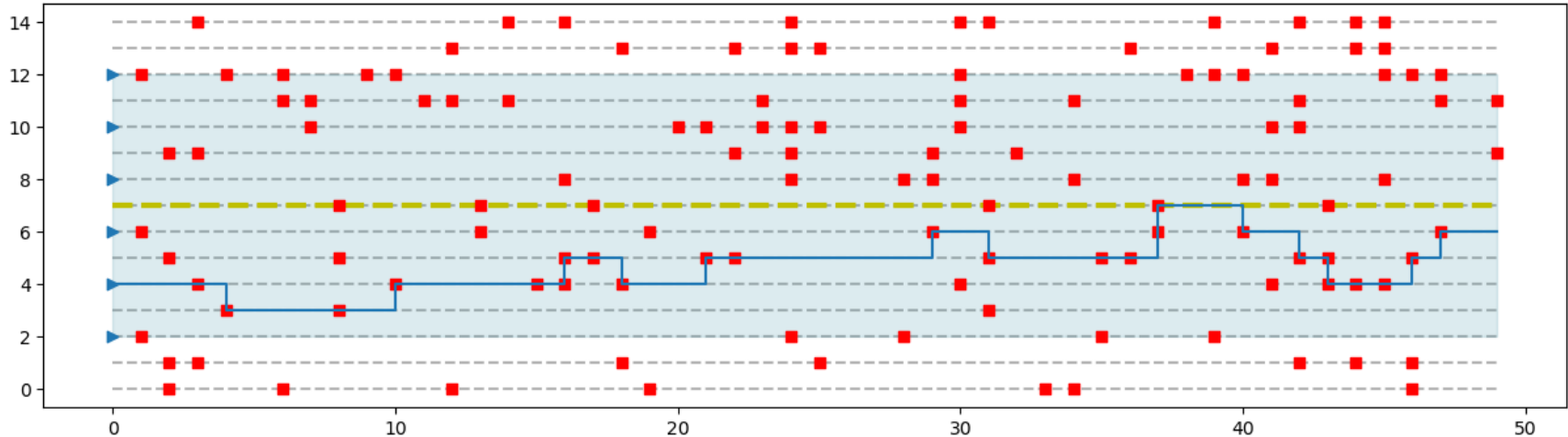


Road Map: avoiding obstacles_Quantum agent

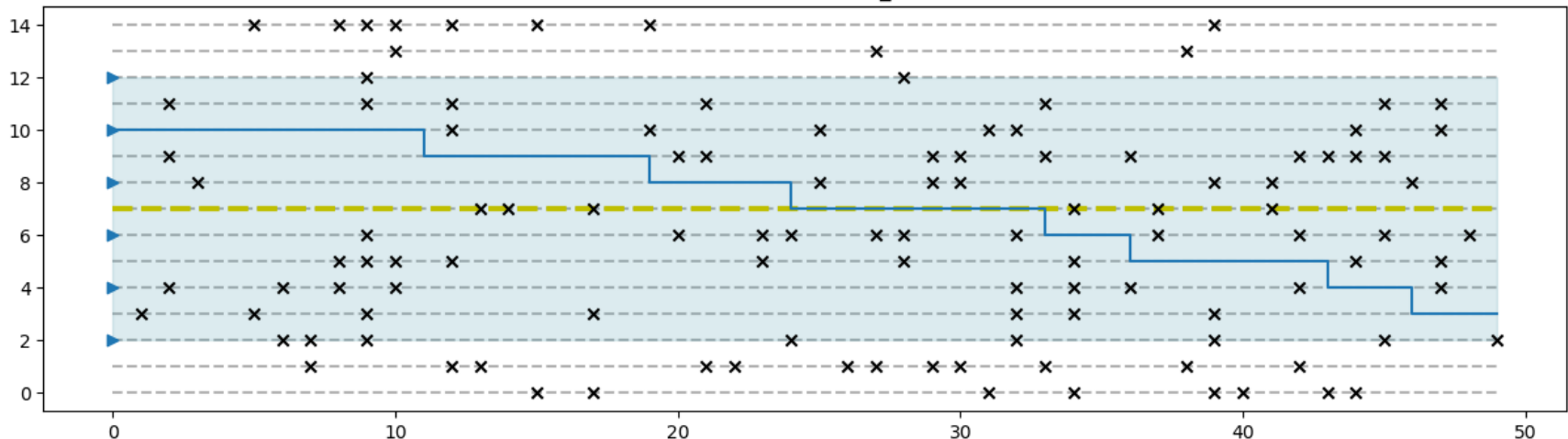


Classical Agent

Road Map: picking up litters_Classical agent



Road Map: avoiding obstacles_Classical agent



Summary and future directions

- We have demonstrated the feasibility of using quantum algorithm to do reinforcement learning for both global (Frozen-lake) and local (Sidewalk) environments.
- For systems with small state space, the quantum algorithm cannot outperform its classical counterpart. More powerful and efficient quantum algorithm is needed to achieve the quantum advantage.
- The quantum agent was implemented on a classical quantum simulator. The next step will be testing this algorithm on a real quantum computer.

Thanks!