# Outline

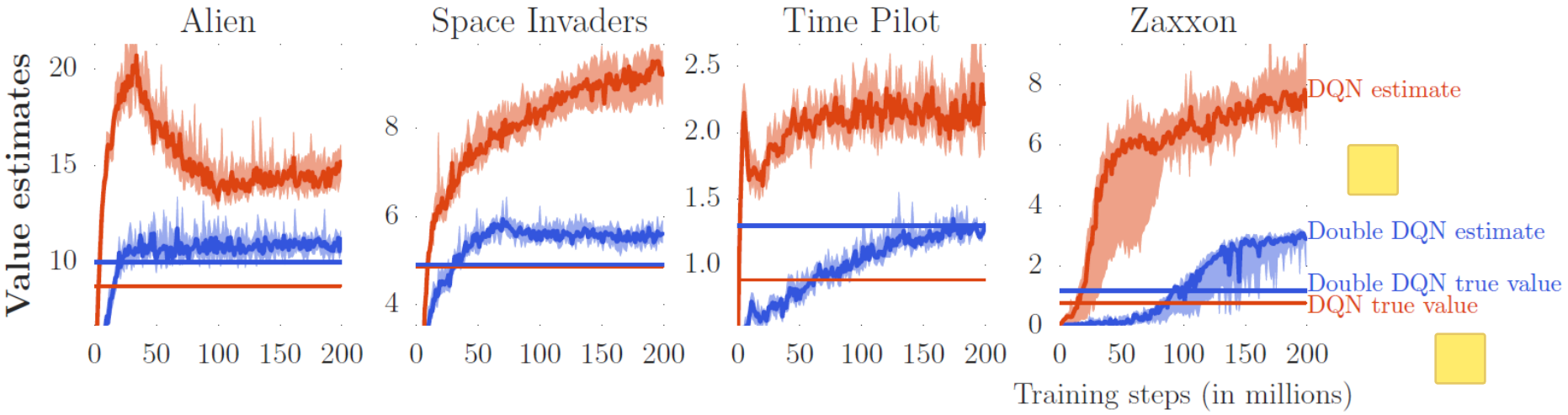
Introduction of Q-Learning

Tips of Q-Learning

Q-Learning for Continuous Actions

# Double DQN

- Q value is usually over-estimated



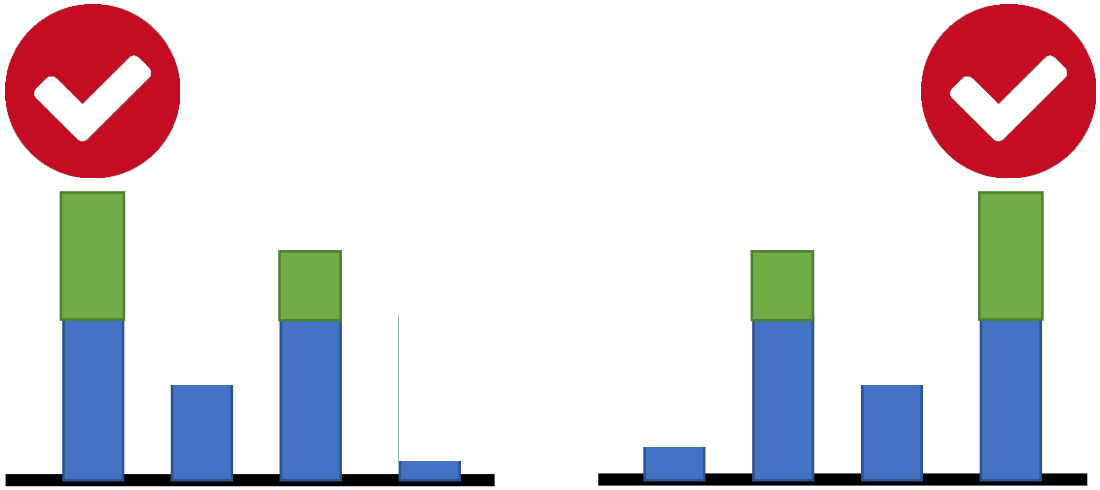
# Double DQN

- Q value is usually over estimate

$$Q(s_t, a_t) \longleftrightarrow r_t + \max_a Q(s_{t+1}, a)$$

Tend to select the action that is over-estimated

$Q(s_{t+1}, a)$



# Double DQN

- Q value is usually over estimate

$$Q(s_t, a_t) \longleftrightarrow r_t + \max_a Q(s_{t+1}, a)$$

- Double DQN: two functions Q and Q' Target Network

$$Q(s_t, a_t) \longleftrightarrow r_t + Q'\left(s_{t+1}, arg\max_a Q(s_{t+1}, a)\right)$$

If Q over-estimate a, so it is selected. Q' would give it proper value.

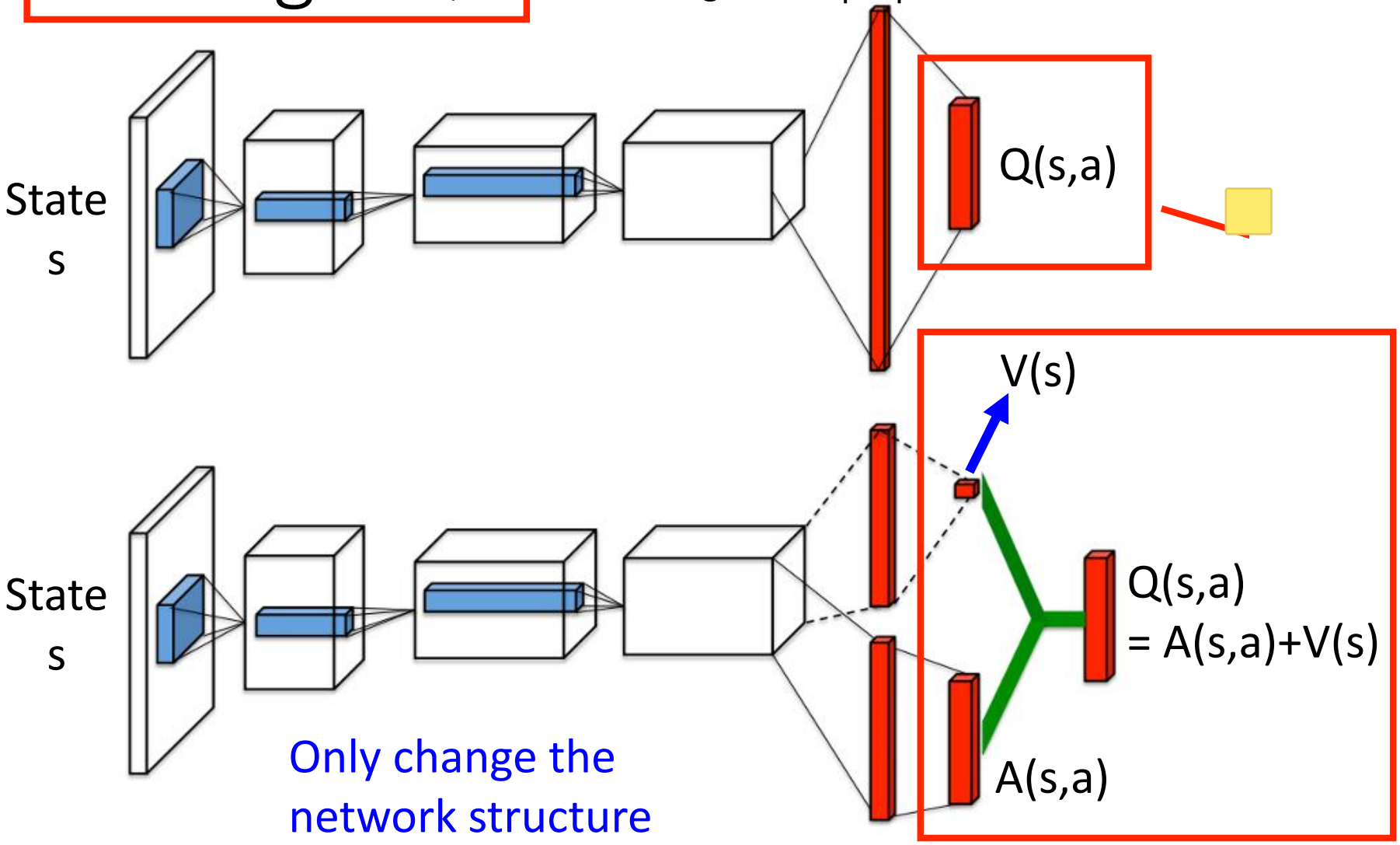
How about Q' overestimate? The action will not be selected by Q.

Hado V. Hasselt, "Double Q-learning", NIPS 2010
Hado van Hasselt, Arthur Guez, David Silver, "Deep Reinforcement Learning with Double Q-learning", AAAI 2016

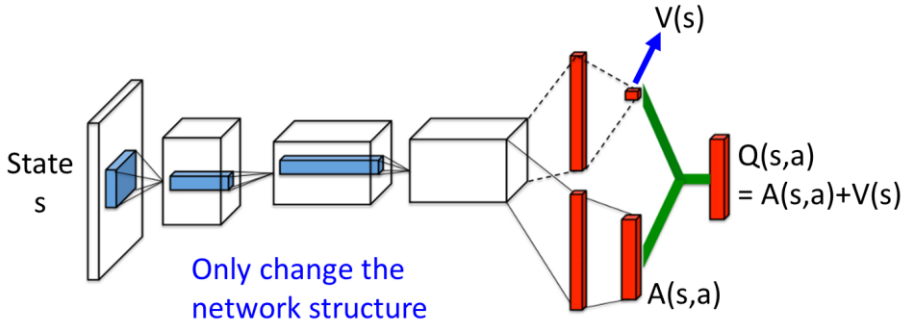
只改變原來 Deep Q Network 的架構

# Dueling DQN

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, Nando de Freitas, "Dueling Network Architectures for Deep Reinforcement Learning", arXiv preprint, 2015



# Dueling DQN



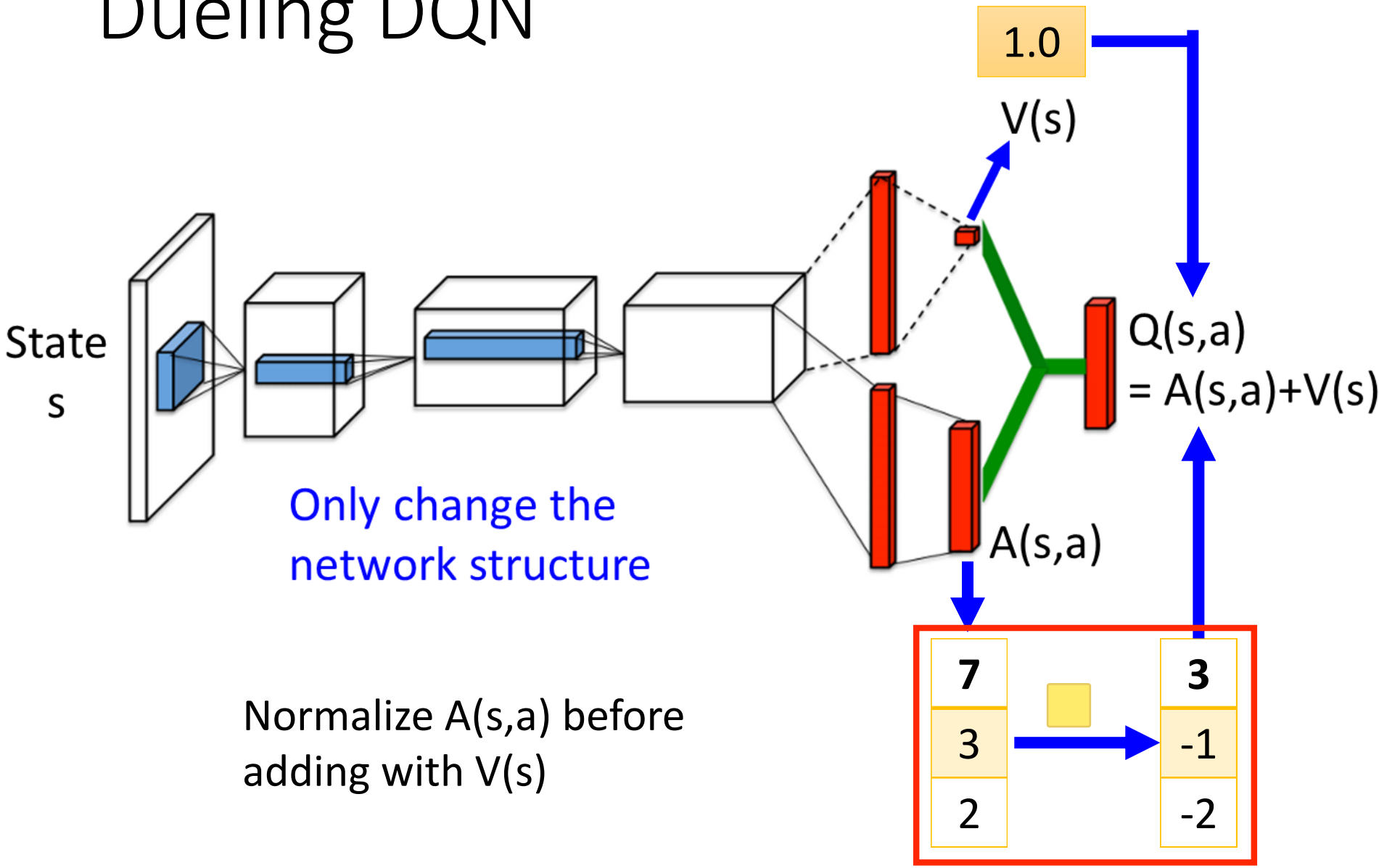

Q(s,a) = V(s) + A(s,a)

| action | state | | | |
|---|---|---|---|---|
| | 3 | ~~3~~ 4 | 3 | 1 |
| | 1 | ~~-1~~ 0 | 6 | 1 |
| | 2 | ~~-2~~ -1 | 3 | 1 |
| | 2 | ~~0~~ 1 | 4 | 1 |
| | 1 | 3 | -1 | 0 |
| | -1 | -1 | 2 | 0 |
| | 0 | -2 | -1 | 0 |

Average of column

sum of column = 0

# Dueling DQN

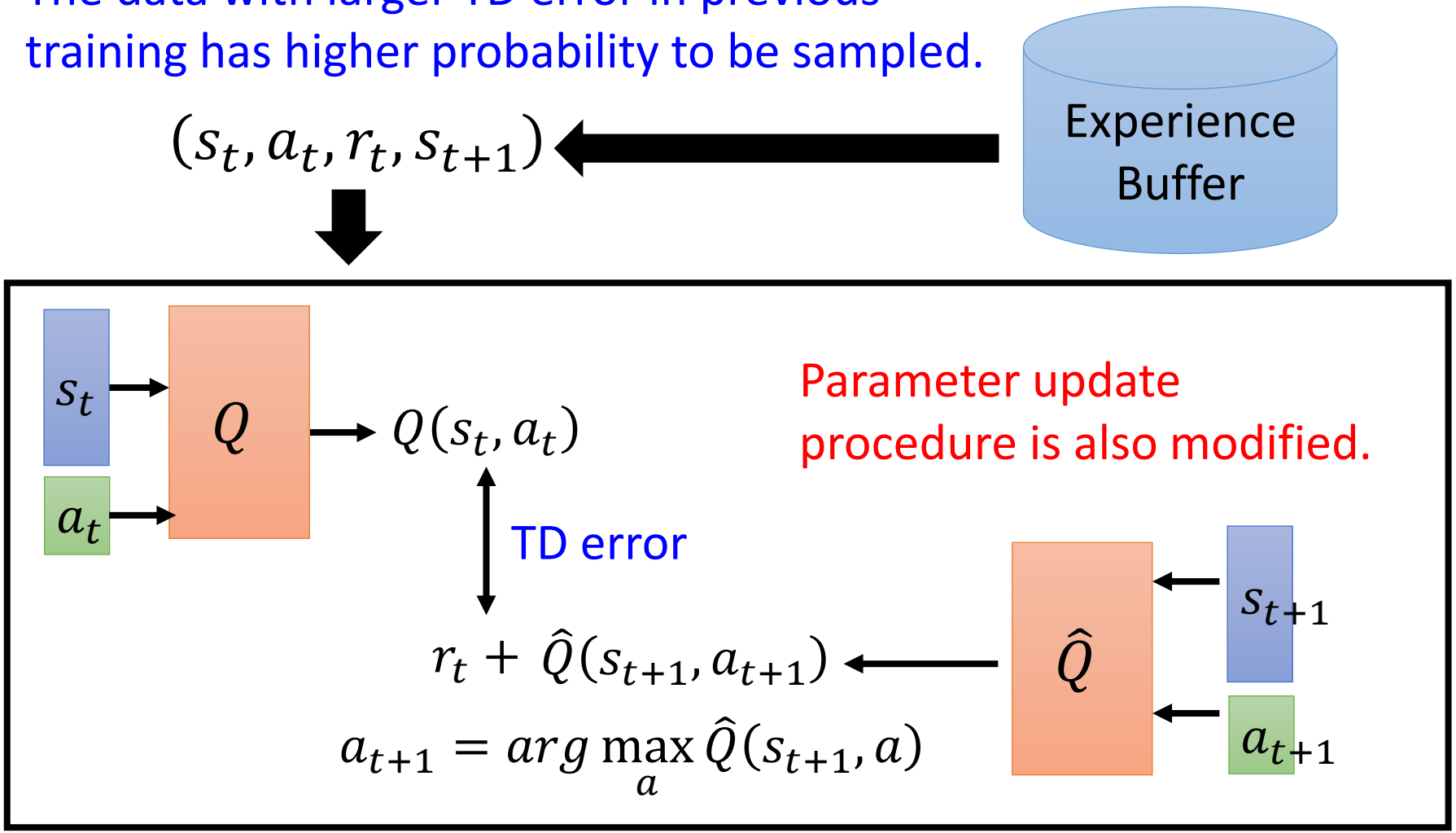



Normalize A(s,a) before adding with V(s)

https://arxiv.org/abs/1511.05952?context=cs

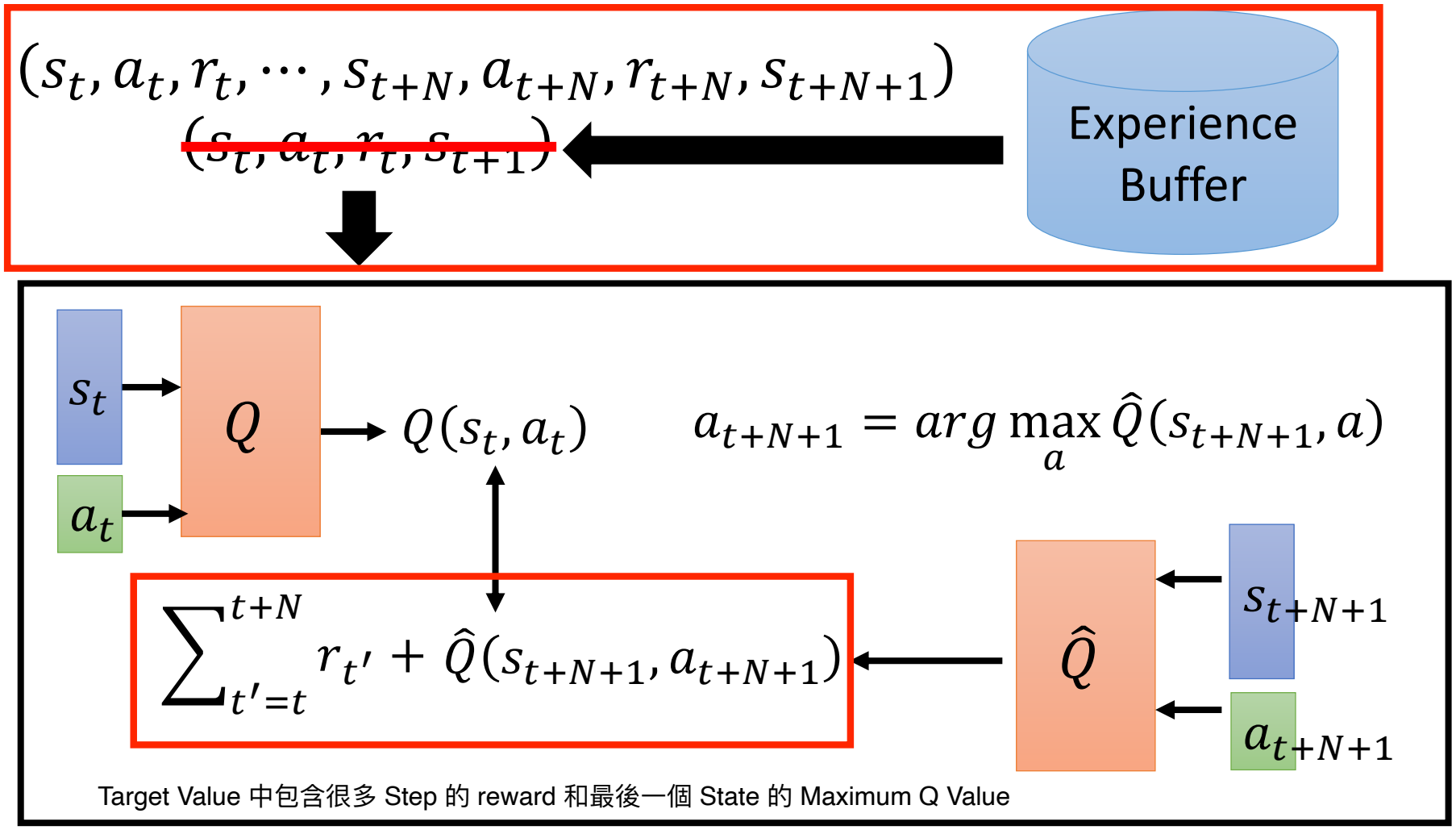
# Prioritized Reply

The data with larger TD error in previous training has higher probability to be sampled.



# Multi-step

Balance between MC and TD



# Noisy Net

https://arxiv.org/abs/1706.01905
https://arxiv.org/abs/1706.10295

- Noise on Action (Epsilon Greedy)

$$a = \begin{cases} arg\max_a Q(s,a), & with\ probability\ 1-\varepsilon \\ random, & otherwise \end{cases}$$

- Noise on Parameters

$$a = arg\max_a \tilde{Q}(s,a)$$

Inject noise into the parameters of Q-function **at the beginning of each episode**

$$Q(s,a) \xrightarrow{\text{Add noise}} \tilde{Q}(s,a)$$

The noise would **NOT** change in an episode.

# Noisy Net

- Noise on Action
  - Given the same state, the agent may takes different actions.
  - No real policy works in this way

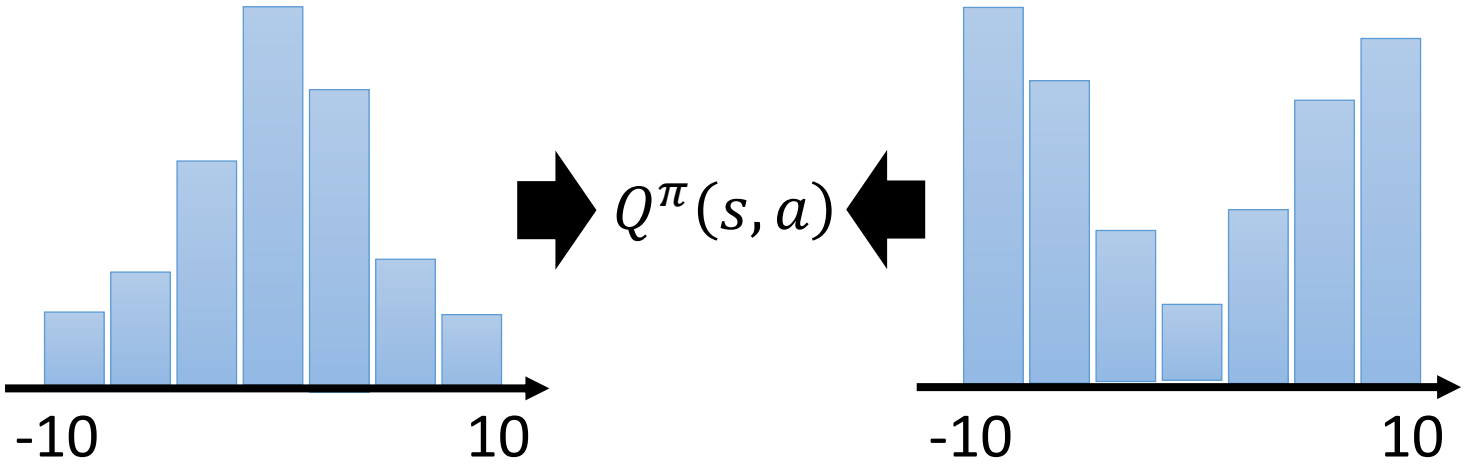
隨機亂試

- Noise on Parameters
  - Given the same (similar) state, the agent takes the same action.
    - ⟶ State-dependent Exploration
  - Explore in a *consistent* way

有系統地試

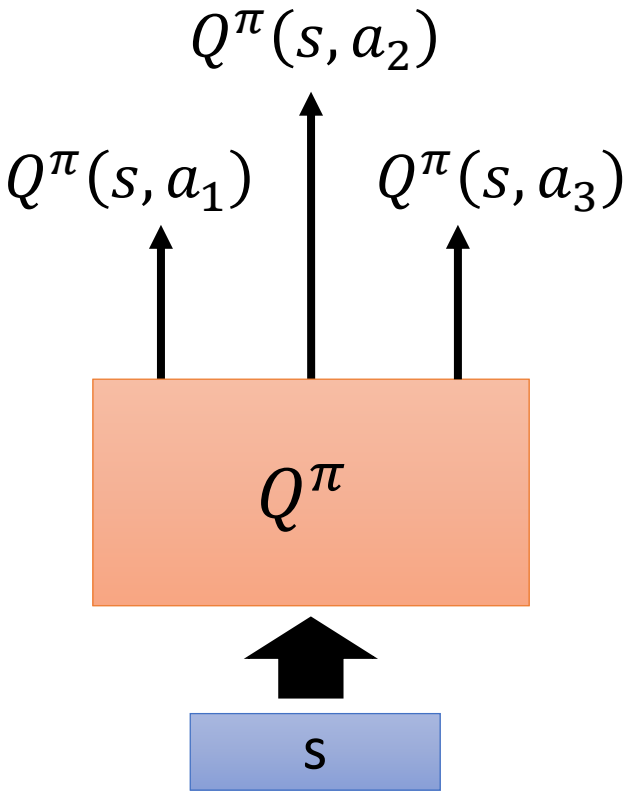
# Distributional Q-function

- State-action value function $Q^{\pi}(s, a)$
  - When using actor $\pi$, the *cumulated* reward expects to be obtained after seeing observation s and taking a

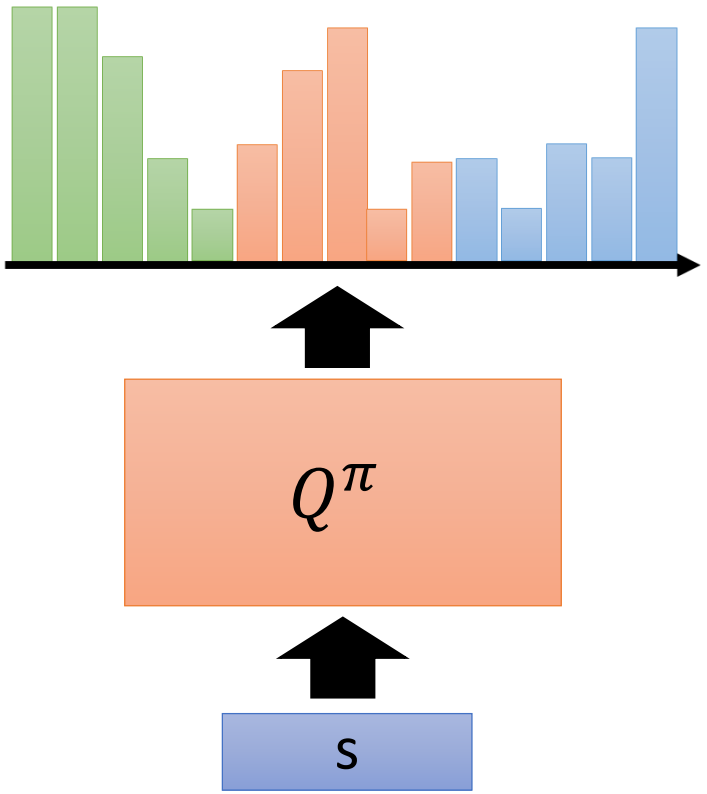



Different distributions can have the same values.

# Distributional Q-function

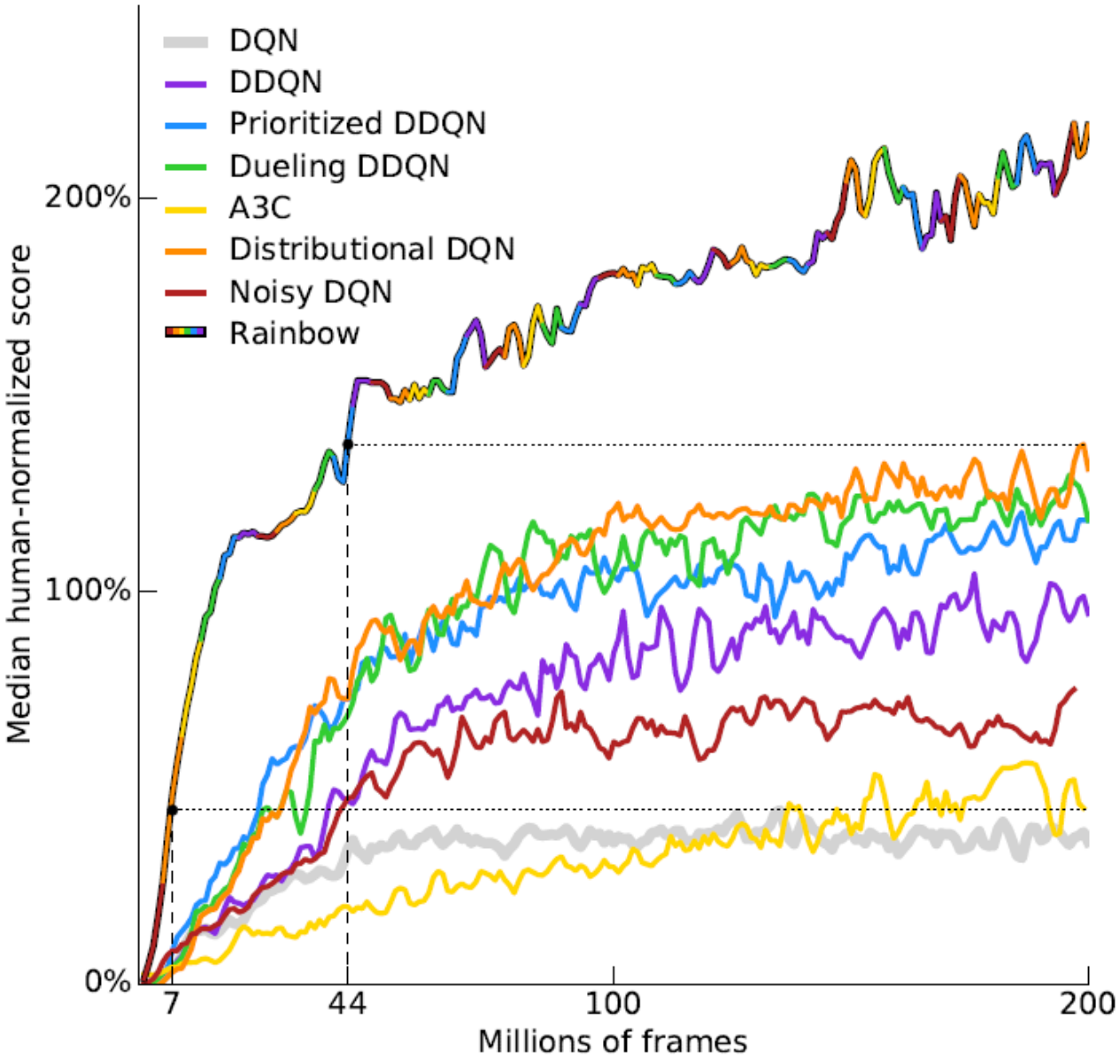



A network with 3 outputs



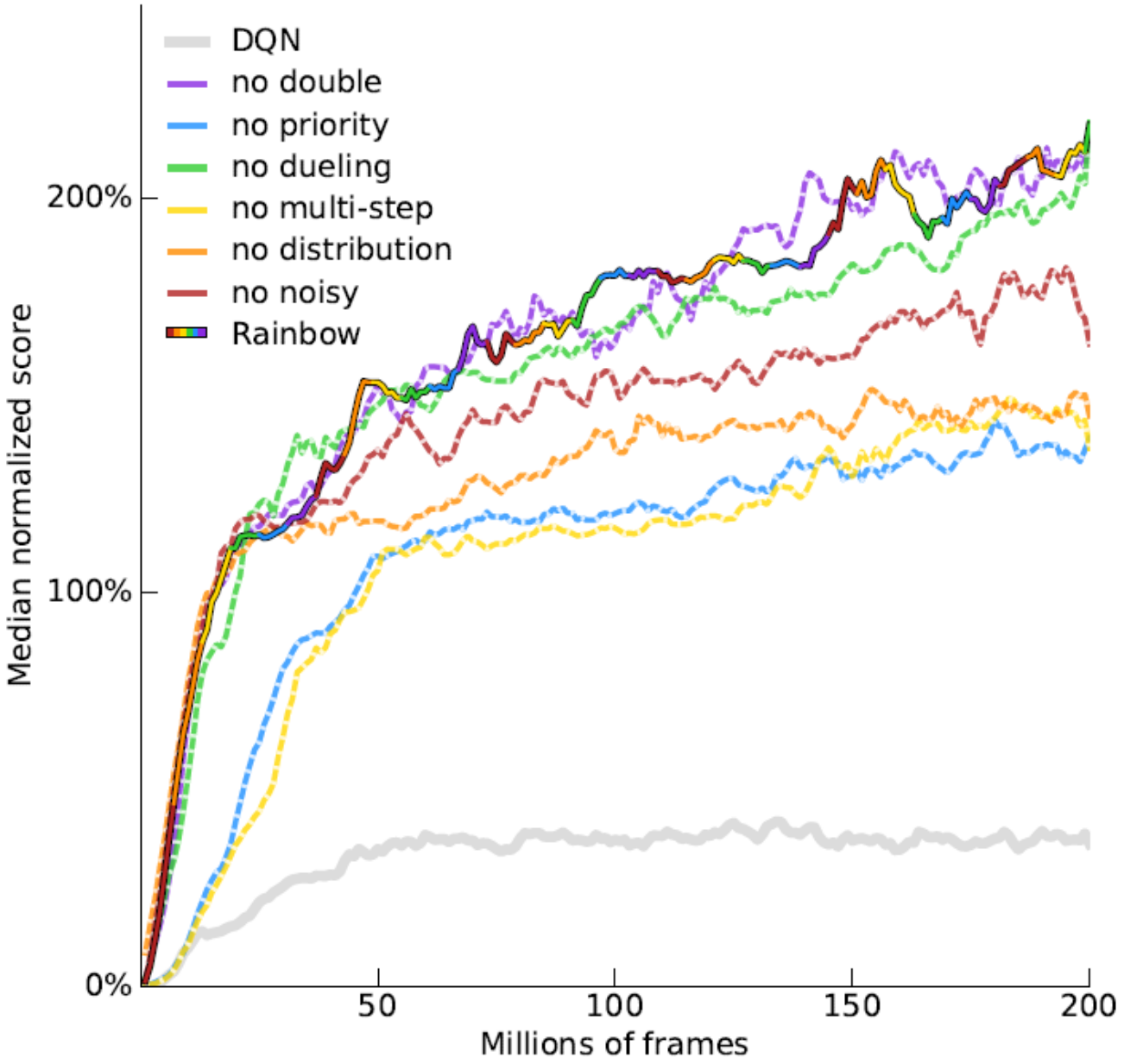

A network with 15 outputs
(each action has 5 bins)

# Rainbow




https://arxiv.org/abs/1710.02298

# Rainbow




https://arxiv.org/abs/1710.02298