AMII TEA TIME TALKS 2020

- This talk will be recorded. If you want to remain anonymous, please join the meeting from an incognito window and keep your video off.

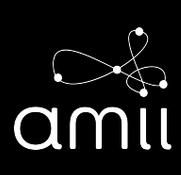
- Clarifications can be asked for right away. There is a Q-A session at the end of the talk for questions of more technical or philosophical kind.

LEARNING AND PLANNING IN AVERAGE-REWARD MDPS

Abhishek Naik

abhishek.naik@ualberta.ca

w/ Yi Wan and Rich Sutton



UNIVERSITY OF ALBERTA

OUTLINE

- Contributions
- Background
  - Problem setting
  - Related work
- Algorithms and Experiments
  - Control
  - Prediction
  - Centering
- Conclusions and Future Work

CONTRIBUTIONS

CONTRIBUTIONS

1. The first general proven-convergent off-policy model-free *control* algorithm without reference states

CONTRIBUTIONS

1. The first general proven-convergent off-policy model-free *control* algorithm without reference states
2. The first proven-convergent off-policy model-free *prediction* algorithm

CONTRIBUTIONS

1. The first general proven-convergent off-policy model-free *control* algorithm without reference states

2. The first proven-convergent off-policy model-free *prediction* algorithm

3. A general technique to estimate the actual *centered* value function rather than the value function plus an offset

PROBLEM SETTING

BACKGROUND

BACKGROUND

- Continuing problems

PROBLEM SETTING

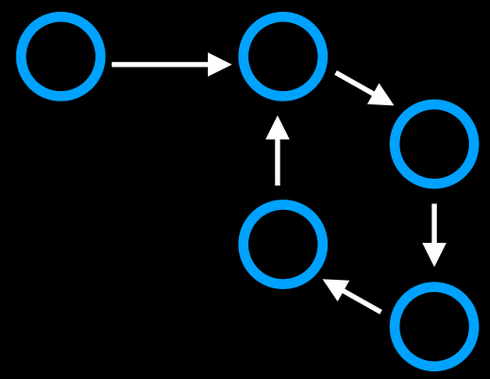
BACKGROUND

- Continuing problems
- Tabular representation

BACKGROUND

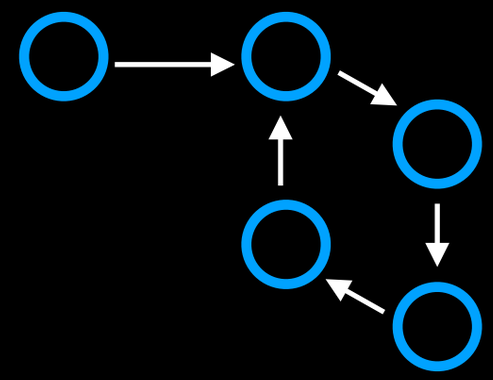
- Continuing problems
- Tabular representation
- Unichain MDPs

BACKGROUND



- Continuing problems
- Tabular representation
- Unichain MDPs

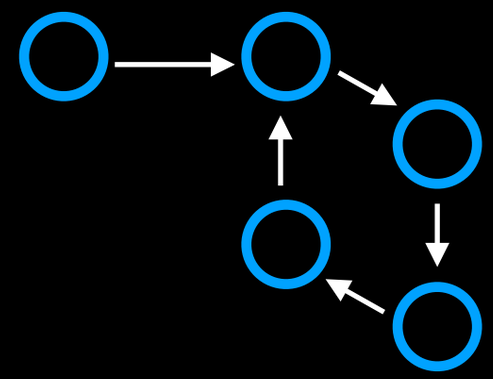
BACKGROUND



- Continuing problems
- Tabular representation
- Unichain MDPs

Reward rate $$r(\pi) = \lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} \mathbb{E}[R_t \,|\, S_0, A_{0:t-1} \sim \pi]$$

BACKGROUND



- Continuing problems
- Tabular representation
- Unichain MDPs

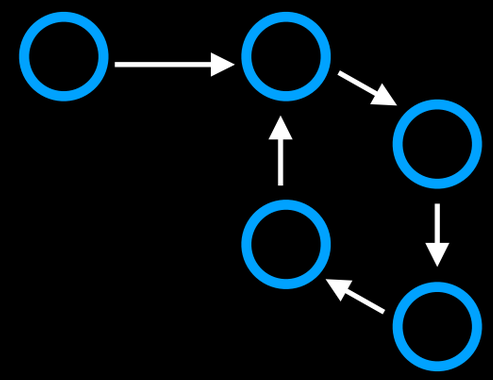
Reward rate

$$r(\pi) = \lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} \mathbb{E}[R_t \,|\, S_0, A_{0:t-1} \sim \pi]$$

Differential value function

$$v_\pi(s) = \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \sum_{t=1}^{k} \mathbb{E}[R_t - r(\pi) \,|\, S_0 = s, A_{0:t-1} \sim \pi] \quad \forall s$$

BACKGROUND



- Continuing problems
- Tabular representation
- Unichain MDPs

Reward rate

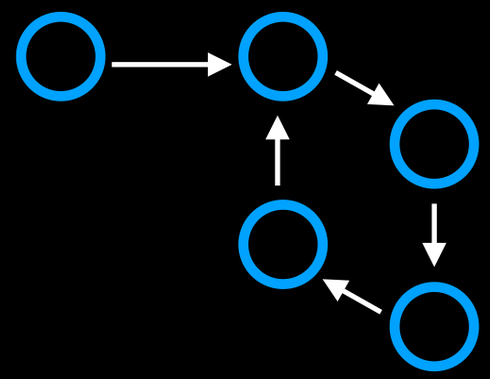
$$r(\pi) = \lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} \mathbb{E}[R_t \,|\, S_0, A_{0:t-1} \sim \pi]$$

Differential value function

$$v_\pi(s) = \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \sum_{t=1}^{k} \mathbb{E}[R_t - r(\pi) \,|\, S_0 = s, A_{0:t-1} \sim \pi] \quad \forall s$$

$$v(s) = \sum_a \pi(a \,|\, s) \sum_{s',r} p(s', r \,|\, s, a) \left[r - \bar{r} + v(s') \right] \quad \forall s$$

BACKGROUND



- Continuing problems
- Tabular representation
- Unichain MDPs

Reward rate

$$r(\pi) = \lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} \mathbb{E}[R_t \,|\, S_0, A_{0:t-1} \sim \pi]$$

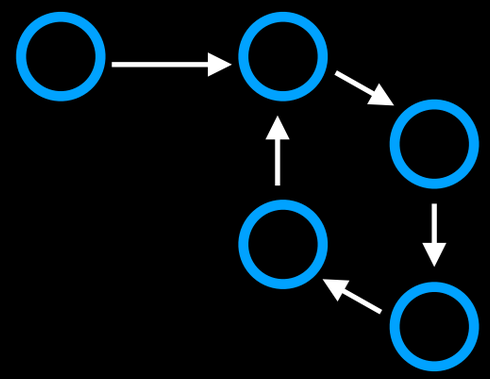
Differential value function

$$v_\pi(s) = \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \sum_{t=1}^{k} \mathbb{E}[R_t - r(\pi) \,|\, S_0 = s, A_{0:t-1} \sim \pi] \quad \forall s$$

$$v(s) = \sum_a \pi(a \,|\, s) \sum_{s',r} p(s', r \,|\, s, a) \big[r - \bar{r} + v(s')\big] \quad \forall s$$

$$q(s, a) = \sum_{s',r} p(s', r \,|\, s, a) \big[r - \bar{r} + \max_{a'} q(s', a')\big] \quad \forall s, a$$

BACKGROUND



- Continuing problems
- Tabular representation
- Unichain MDPs

Reward rate

$$r(\pi) = \lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} \mathbb{E}[R_t \,|\, S_0, A_{0:t-1} \sim \pi]$$

Differential value function

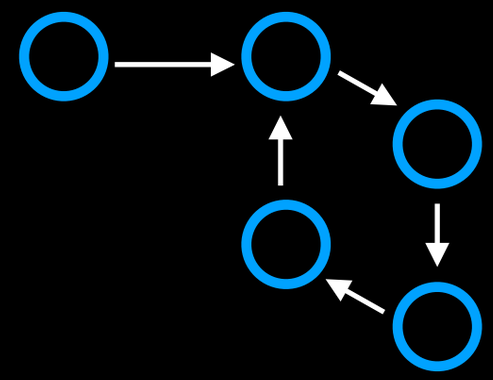
$$v_\pi(s) = \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \sum_{t=1}^{k} \mathbb{E}[R_t - r(\pi) \,|\, S_0 = s, A_{0:t-1} \sim \pi] \quad \forall s$$

Bellman equations

$$v(s) = \sum_a \pi(a \,|\, s) \sum_{s',r} p(s', r \,|\, s, a) \left[r - \bar{r} + v(s') \right] \quad \forall s$$

$$q(s,a) = \sum_{s',r} p(s', r \,|\, s, a) \left[r - \bar{r} + \max_{a'} q(s', a') \right] \quad \forall s, a$$

BACKGROUND



- Continuing problems
- Tabular representation
- Unichain MDPs

Unique solution for $\bar{r}$,
multiple solutions for v, q

Reward rate

$$r(\pi) = \lim_{n\to\infty} \frac{1}{n} \sum_{t=1}^{n} \mathbb{E}[R_t \,|\, S_0, A_{0:t-1} \sim \pi]$$

Differential value function

$$v_\pi(s) = \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} \sum_{t=1}^{k} \mathbb{E}[R_t - r(\pi) \,|\, S_0 = s, A_{0:t-1} \sim \pi] \quad \forall s$$

Bellman equations

$$v(s) = \sum_a \pi(a \,|\, s) \sum_{s',r} p(s', r \,|\, s, a) \left[r - \bar{r} + v(s') \right] \quad \forall s$$

$$q(s,a) = \sum_{s',r} p(s', r \,|\, s, a) \left[r - \bar{r} + \max_{a'} q(s', a') \right] \quad \forall s, a$$

BACKGROUND

Average-reward *learning* algorithms	Prediction	Control
On-policy		
Off-policy		

BACKGROUND

Average-reward *learning* algorithms	**Prediction**	**Control**
On-policy	Average Cost TD (1999)	Actor-critic (2000, 2009)
Off-policy		R-learning (1993) Singh (1994) RVI Q-learning (2001) Gosavi (2004)

BACKGROUND

Average-reward *learning* algorithms	**Prediction**	**Control**
On-policy	Average Cost TD (1999)	Actor-critic (2000, 2009)
Off-policy	*Differential TD-learning*	R-learning (1993) Singh (1994) RVI Q-learning (2001) Gosavi (2004) *Differential Q-learning*

BACKGROUND

Average-reward *learning* algorithms	**Prediction**	**Control**
On-policy	Average Cost TD (1999)	Actor-critic (2000, 2009)
Off-policy	*Differential TD-learning*	R-learning (1993) Singh (1994) RVI Q-learning (2001) Gosavi (2004) *Differential Q-learning*

Many algorithms that minimize regret
(UCRL2, POLITEX, Opt-QL, EE-QL)

BACKGROUND

Average-reward *learning* algorithms	**Prediction**	**Control**
On-policy	Average Cost TD (1999)	Actor-critic (2000, 2009)
Off-policy	*Differential TD-learning*	R-learning (1993) Singh (1994) RVI Q-learning (2001) Gosavi (2004) *Differential Q-learning*

Many algorithms that minimize regret, but: typically not off-policy as a behavioural policy needs to be specified,
(UCRL2, POLITEX, Opt-QL, EE-QL) value estimates might be unbounded.

BACKGROUND

BACKGROUND

$$v(s) = \sum_{a} \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a) \left[r - \bar{r} + v(s') \right] \quad \forall s$$

BACKGROUND

$$v(s) = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a) \left[r - \bar{r} + v(s') \right] \quad \forall s$$

$$\bar{r} = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a) \left[r - v(s) + v(s') \right] \quad \forall s$$

BACKGROUND

$$v(s) = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a) \left[r - \bar{r} + v(s') \right] \quad \forall s$$

$$\bar{r} = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a) \left[r - v(s) + v(s') \right] \quad \forall s$$

$$\bar{R}_{t+1} = \bar{R}_t + \beta \left(R_{t+1} - V(S_t) + V(S_{t+1}) - \bar{R}_t \right)$$

BACKGROUND

$$v(s) = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a) \left[r - \bar{r} + v(s') \right] \quad \forall s$$

$$\bar{r} = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a) \left[r - v(s) + v(s') \right] \quad \forall s$$

$$\bar{R}_{t+1} = \bar{R}_t + \beta \left(R_{t+1} - V(S_t) + V(S_{t+1}) - \bar{R}_t \right)$$

$$r(\pi) = \sum_s d_\pi(s) \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a)\, r$$

BACKGROUND

$$v(s) = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a) \left[r - \bar{r} + v(s') \right] \quad \forall s$$

$$\bar{r} = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a) \left[r - v(s) + v(s') \right] \quad \forall s$$

$$\bar{R}_{t+1} = \bar{R}_t + \beta \left(R_{t+1} - V(S_t) + V(S_{t+1}) - \bar{R}_t \right)$$

$$r(\pi) = \sum_s d_\pi(s) \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a)\, r$$

$$\bar{R}_{t+1} = \bar{R}_t + \beta (R_{t+1} - \bar{R}_t)$$

BACKGROUND

$$v(s) = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a) \left[r - \bar{r} + v(s') \right] \quad \forall s$$

$$\bar{r} = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a) \left[r - v(s) + v(s') \right] \quad \forall s$$

$$\bar{R}_{t+1} = \bar{R}_t + \beta \left(R_{t+1} - V(S_t) + V(S_{t+1}) - \bar{R}_t \right)$$

$$r(\pi) = \sum_s d_\pi(s) \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a)\, r$$

$$\bar{R}_{t+1} = \bar{R}_t + \beta (R_{t+1} - \bar{R}_t)$$

ALGORITHM

CONTROL

ALGORITHM

CONTROL

$$\delta_t \doteq R_{t+1} - \bar{R}_t + \max_a Q_t(S_{t+1}, a) - Q_t(S_t, A_t)$$

ALGORITHM

CONTROL

$$\delta_t \doteq R_{t+1} - \bar{R}_t + \max_a Q_t(S_{t+1}, a) - Q_t(S_t, A_t)$$

$$Q_{t+1}(S_t, A_t) \doteq Q_t(S_t, A_t) + \alpha_t \delta_t$$

$$\bar{R}_{t+1} \doteq \bar{R}_t + \eta \alpha_t \delta_t$$

ALGORITHM

CONTROL

Differential Q-learning

$$\delta_t \doteq R_{t+1} - \bar{R}_t + \max_a Q_t(S_{t+1}, a) - Q_t(S_t, A_t)$$

$$Q_{t+1}(S_t, A_t) \doteq Q_t(S_t, A_t) + \alpha_t \delta_t$$

$$\bar{R}_{t+1} \doteq \bar{R}_t + \eta \alpha_t \delta_t$$

ALGORITHM

CONTROL

Differential Q-learning

$$\delta_t \doteq R_{t+1} - \bar{R}_t + \max_a Q_t(S_{t+1}, a) - Q_t(S_t, A_t)$$

$$Q_{t+1}(S_t, A_t) \doteq Q_t(S_t, A_t) + \alpha_t \delta_t$$

$$\bar{R}_{t+1} \doteq \bar{R}_t + \eta \alpha_t \delta_t$$

Theorem 1 (sketch)

If 1) the MDP is unichain,
2) the stepsizes are decreased appropriately,
3) all the state-action pairs are updated infinite number of times,
4) the maximum ratio of the update frequencies is finite,

then the Differential Q-learning algorithm converges a.s.:
$\bar{R}_t$ to $r(\pi^*)$, Q_t to a solution of the Bellman optimality equation.

ALGORITHM

CONTROL

Algorithm 1: Differential Q-learning (one-step off-policy control)

Input: The policy b to be used (e.g., ϵ-greedy)
Algorithm parameters: step size α, η
Initialize $Q(s,a)\ \forall s, a$; $\bar{R}$ arbitrarily (e.g., to zero)
Obtain initial S
while *still time to train* **do**
  $A \leftarrow$ action given by b for S
  Take action A, observe R, S'
  $\delta = R - \bar{R} + \max_a Q(S', a) - Q(S, A)$
  $Q(S, A) = Q(S, A) + \alpha\delta$
  $\bar{R} = \bar{R} + \eta\alpha\delta$
  $S = S'$
end
return Q

CONTROL

Algorithm 1: Differential Q-learning (one-step off-policy control)

Input: The policy b to be used (e.g., ϵ-greedy)
Algorithm parameters: step size α, η
Initialize $Q(s,a)\ \forall s, a$; $\bar{R}$ arbitrarily (e.g., to zero)
Obtain initial S
while *still time to train* **do**
  $A \leftarrow$ action given by b for S
  Take action A, observe R, S'
  $\delta = R - \bar{R} + \max_a Q(S', a) - Q(S, A)$
  $Q(S, A) = Q(S, A) + \alpha\delta$
  $\bar{R} = \bar{R} + \eta\alpha\delta$
  $S = S'$
end
return Q

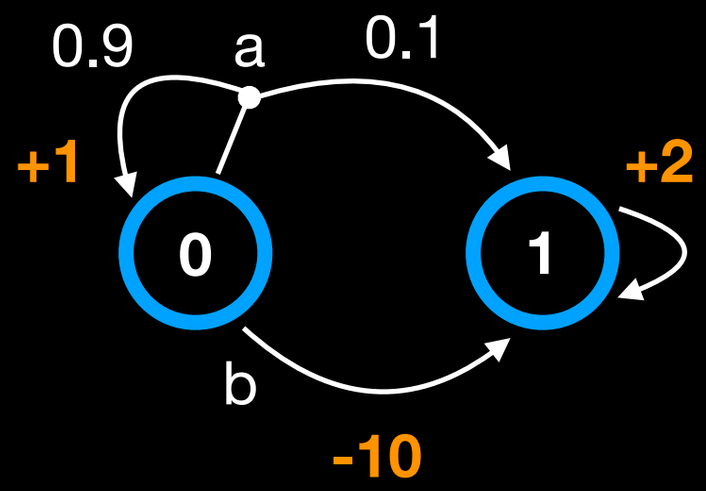
RVI Q-learning

$$Q_{t+1}(S_t, A_t) = Q_t(S_t, A_t) + \alpha_t\big(R_{t+1} - f(Q_t) + \max_a Q_t(S_{t+1}, a) - Q_t(S_t, A_t)\big)$$

CONTROL

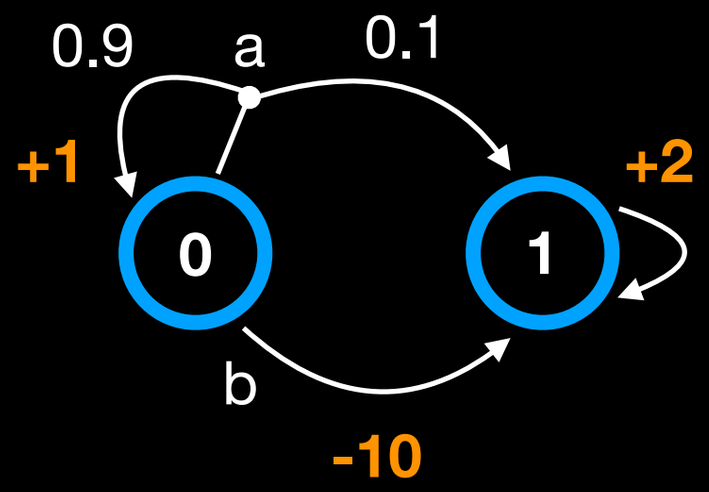
CONTROL

- Two-state MDP
  - state 0 is transient



CONTROL

- Two-state MDP
  - state 0 is transient



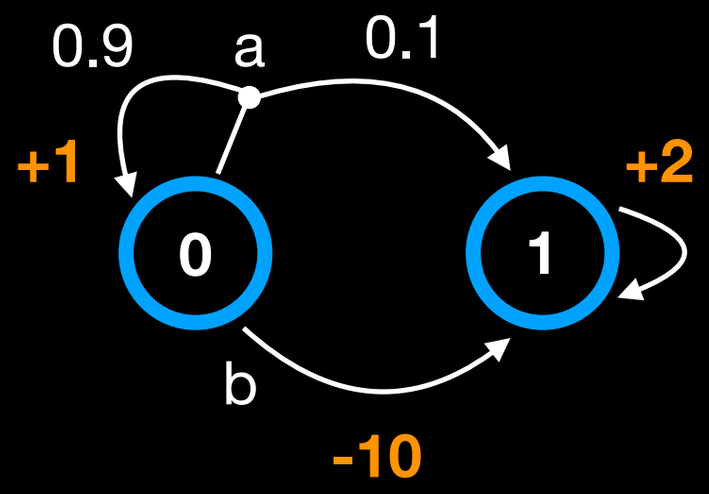
- Reference state-action pair: (0, a)

  i.e., $f(Q) = Q(0, a)$

- $\alpha = 0.01,\ \eta = 1$

CONTROL

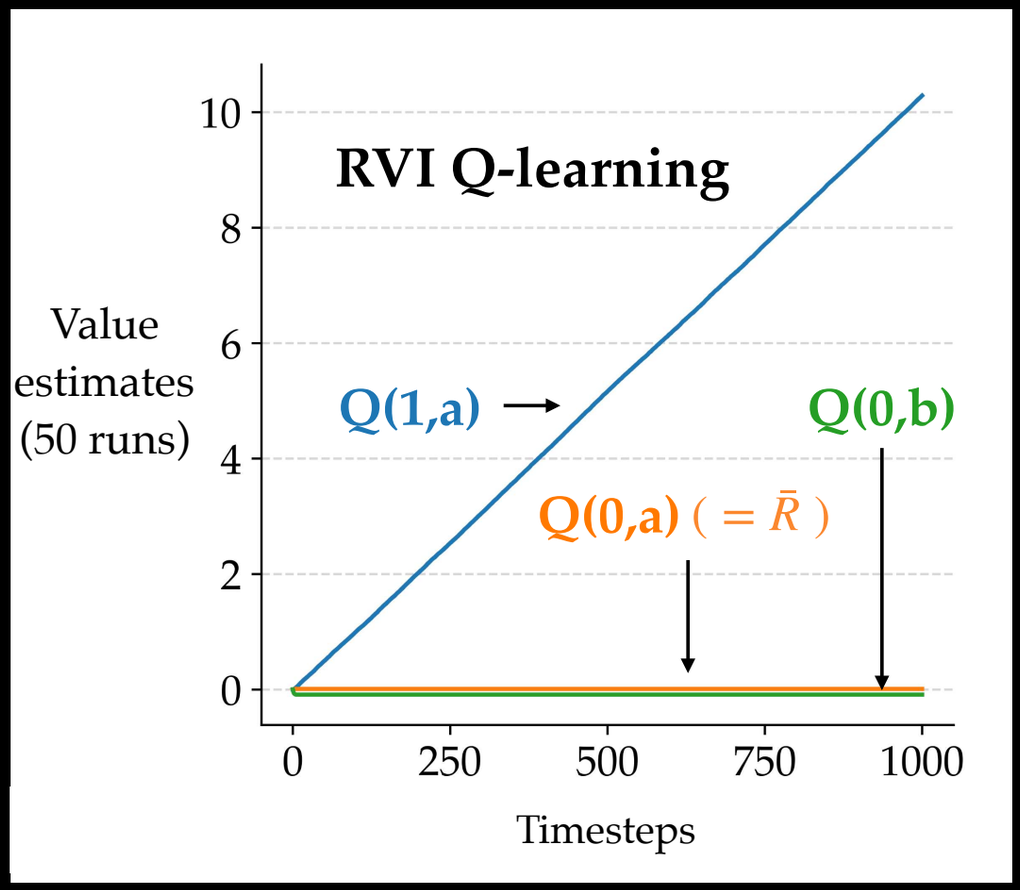
- Two-state MDP
  - state 0 is transient



- Reference state-action pair: (0, a)

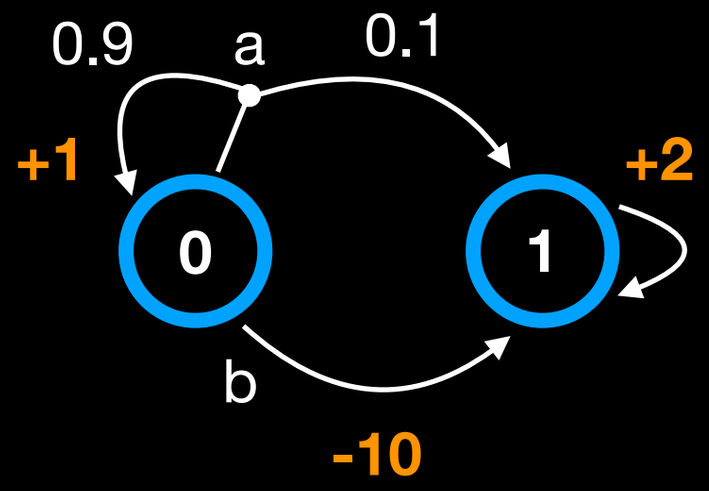
  i.e., $f(Q) = Q(0, a)$

- $\alpha = 0.01,\ \eta = 1$



CONTROL

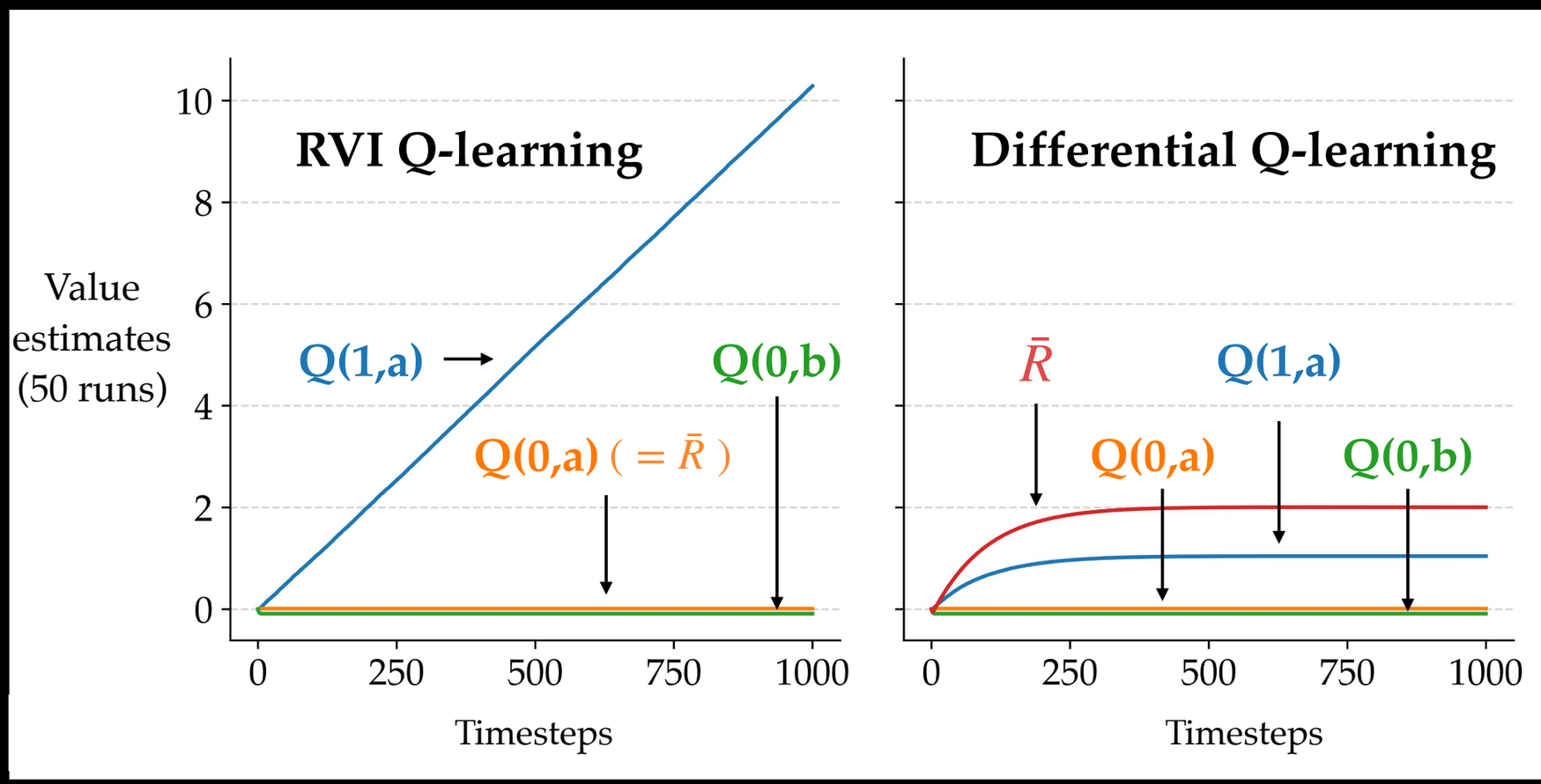
- Two-state MDP
  - state 0 is transient



- Reference state-action pair: (0, a)

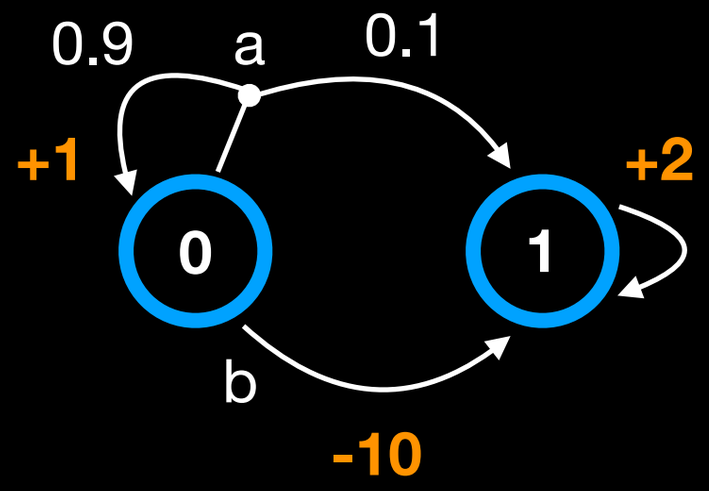
  i.e., $f(Q) = Q(0, a)$

- $\alpha = 0.01,\ \eta = 1$



CONTROL

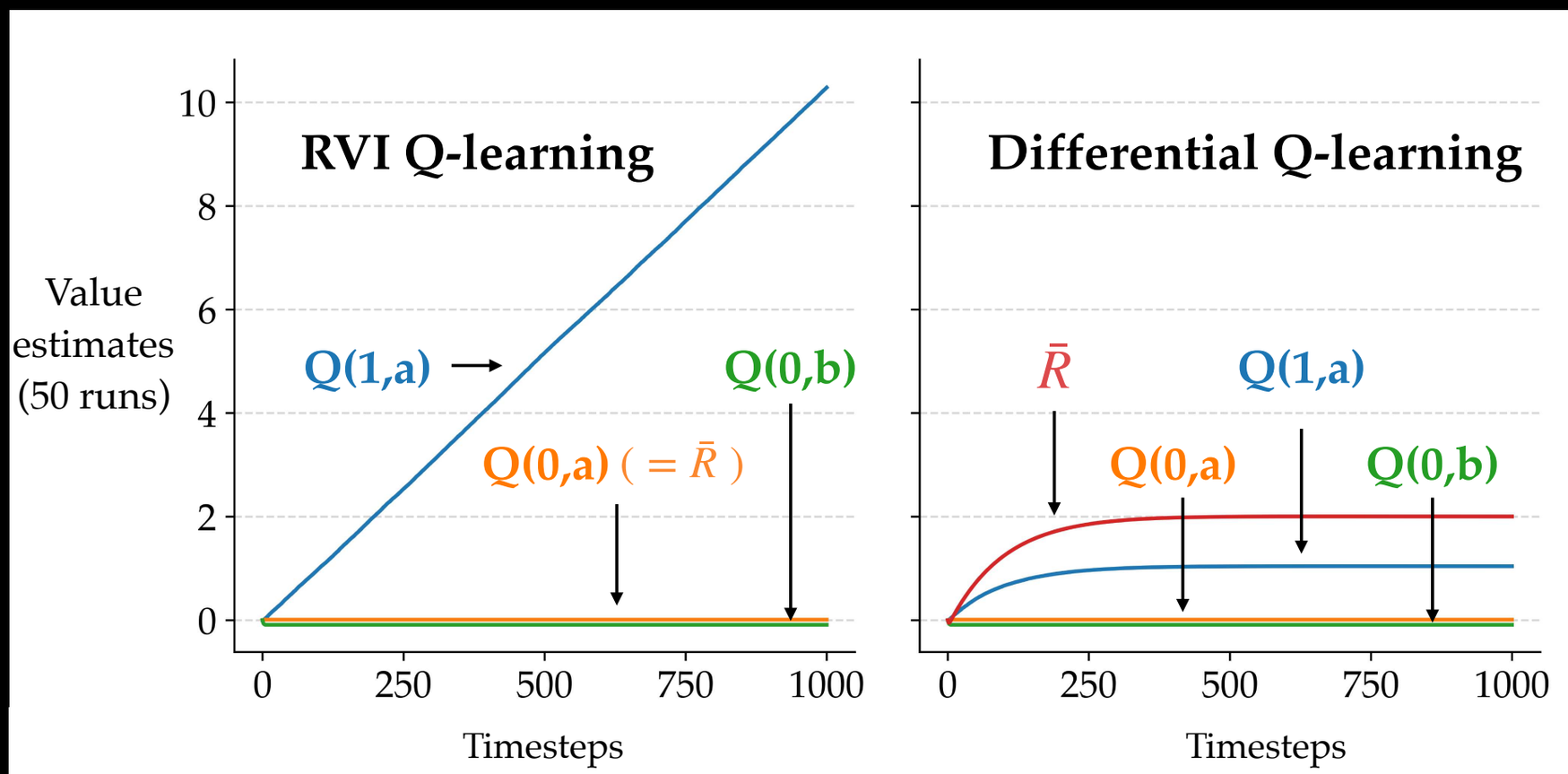
- Two-state MDP
  - state 0 is transient



- Reference state-action pair: (0, a)

  i.e., $f(Q) = Q(0, a)$

- $\alpha = 0.01,\ \eta = 1$

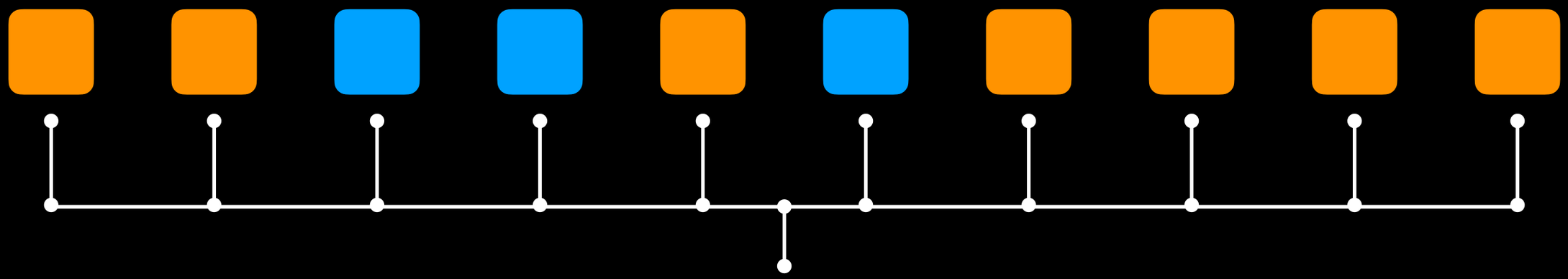


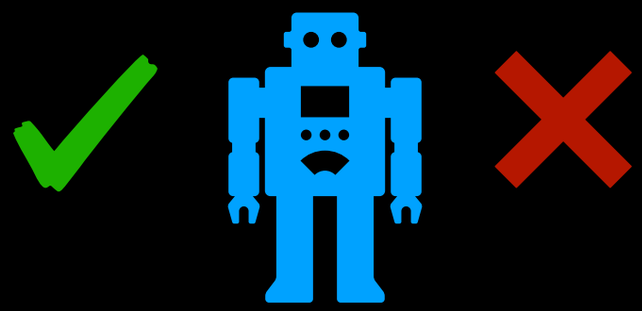
RVI Q-learning diverges if reference state is transient.

ENVIRONMENT

CONTROL

- Access Control Queueing Task



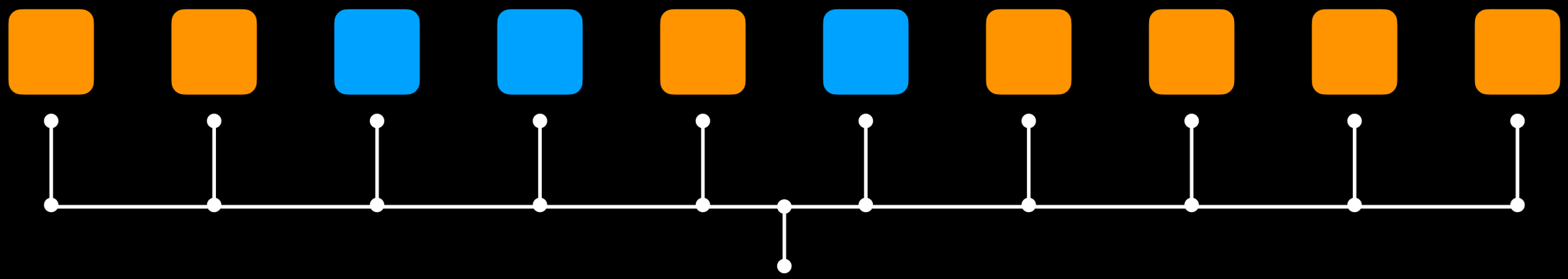


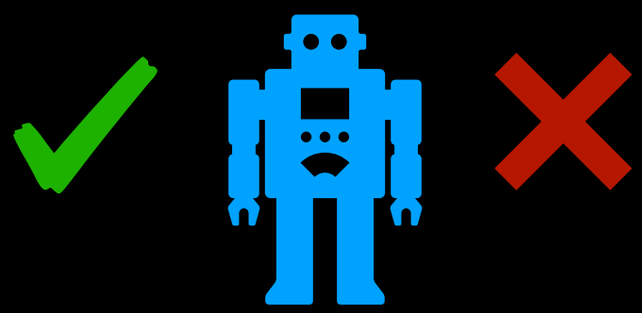
4 8 1 4 2

ENVIRONMENT

CONTROL

- Access Control Queueing Task

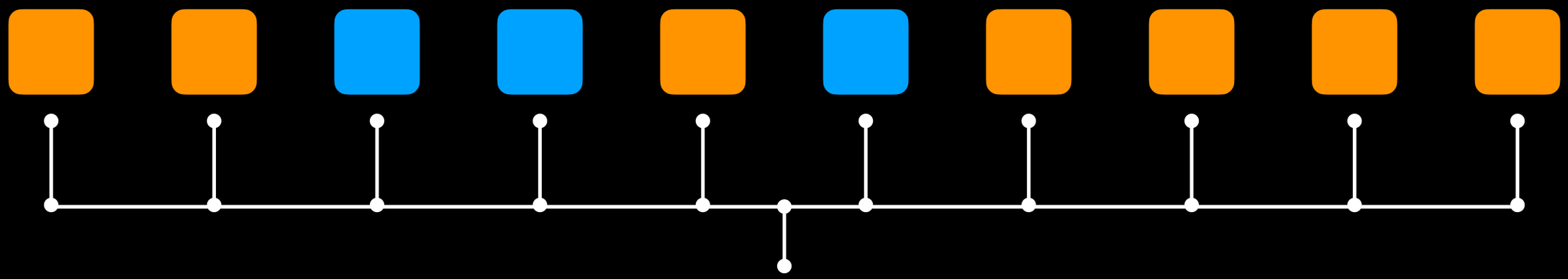


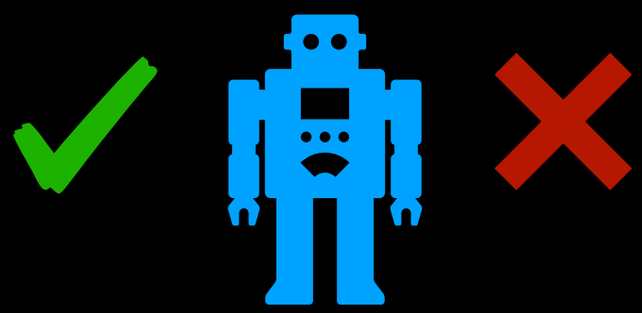


- 10 servers, 4 priorities
- p = 0.06

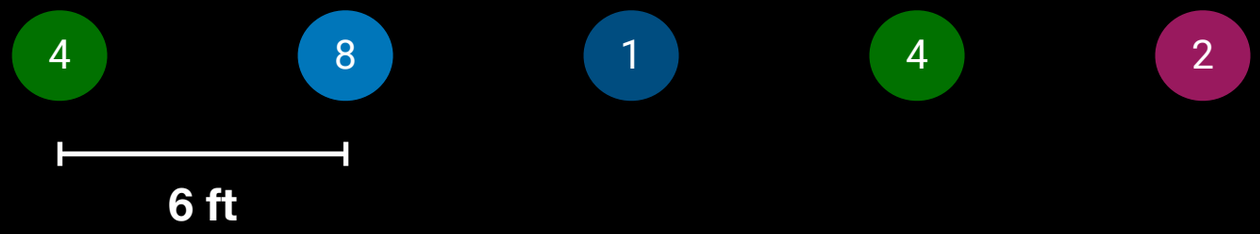
CONTROL

- Access Control Queueing Task





- 10 servers, 4 priorities
- p = 0.06



EXPERIMENT

CONTROL

- $\alpha \in \{0.0015625,\ 0.00625,\ 0.025,\ 0.1,\ 0.4\}$
- $\eta \in \{0.125,\ 0.25,\ 0.5,\ 1,\ 2\}$
- $\epsilon = 0.1$
- 80,000 steps
- 30 runs

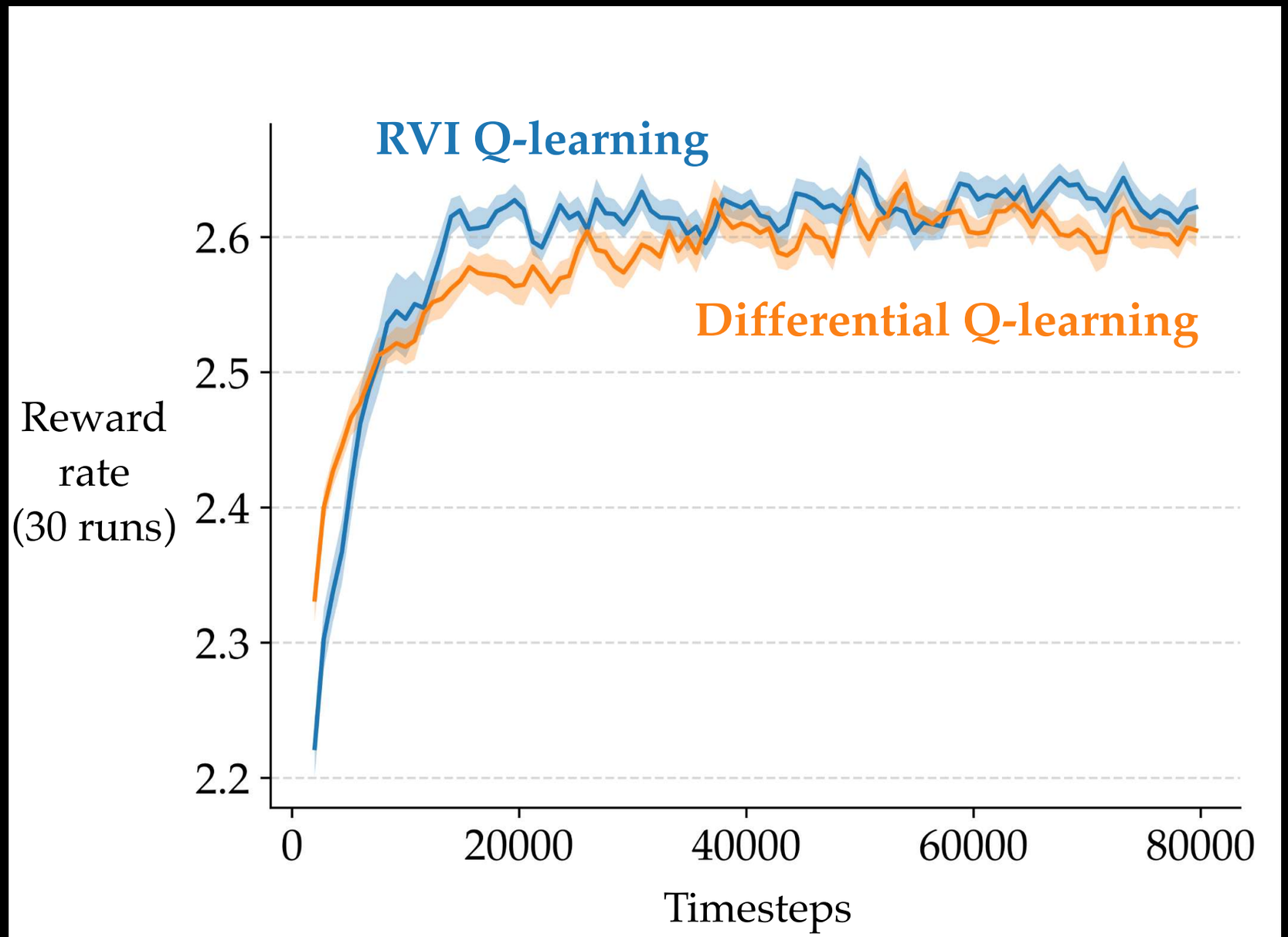
- Reference states:
  - 0, 2, 4, 6, 8, 10 free servers
  - priority 8
  - accept

EXPERIMENT

CONTROL

- $\alpha \in \{0.0015625, 0.00625, 0.025, 0.1, 0.4\}$
- $\eta \in \{0.125, 0.25, 0.5, 1, 2\}$
- $\epsilon = 0.1$
- 80,000 steps
- 30 runs

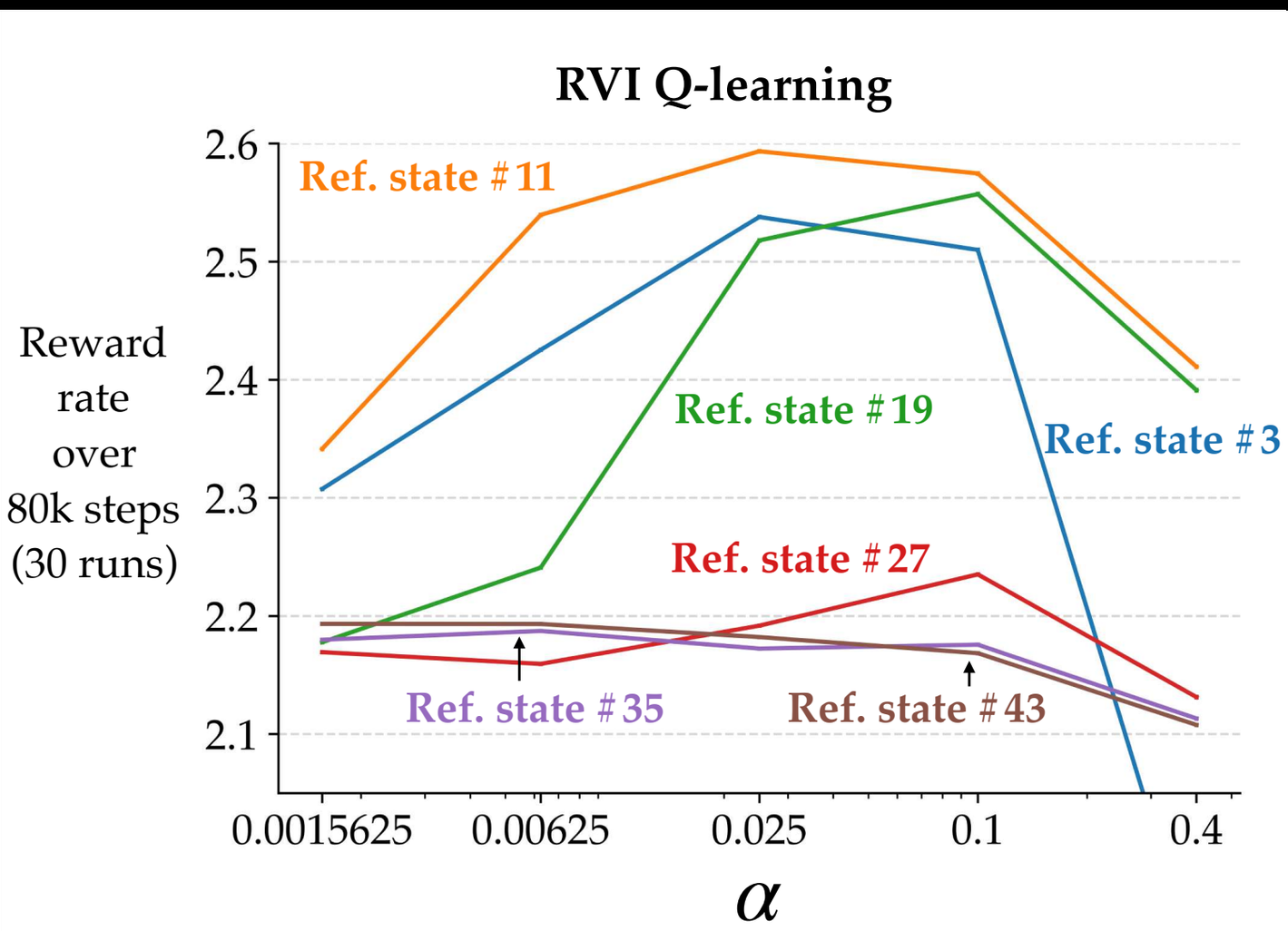
- Reference states:
  - 0, 2, 4, 6, 8, 10 free servers
  - priority 8
  - accept

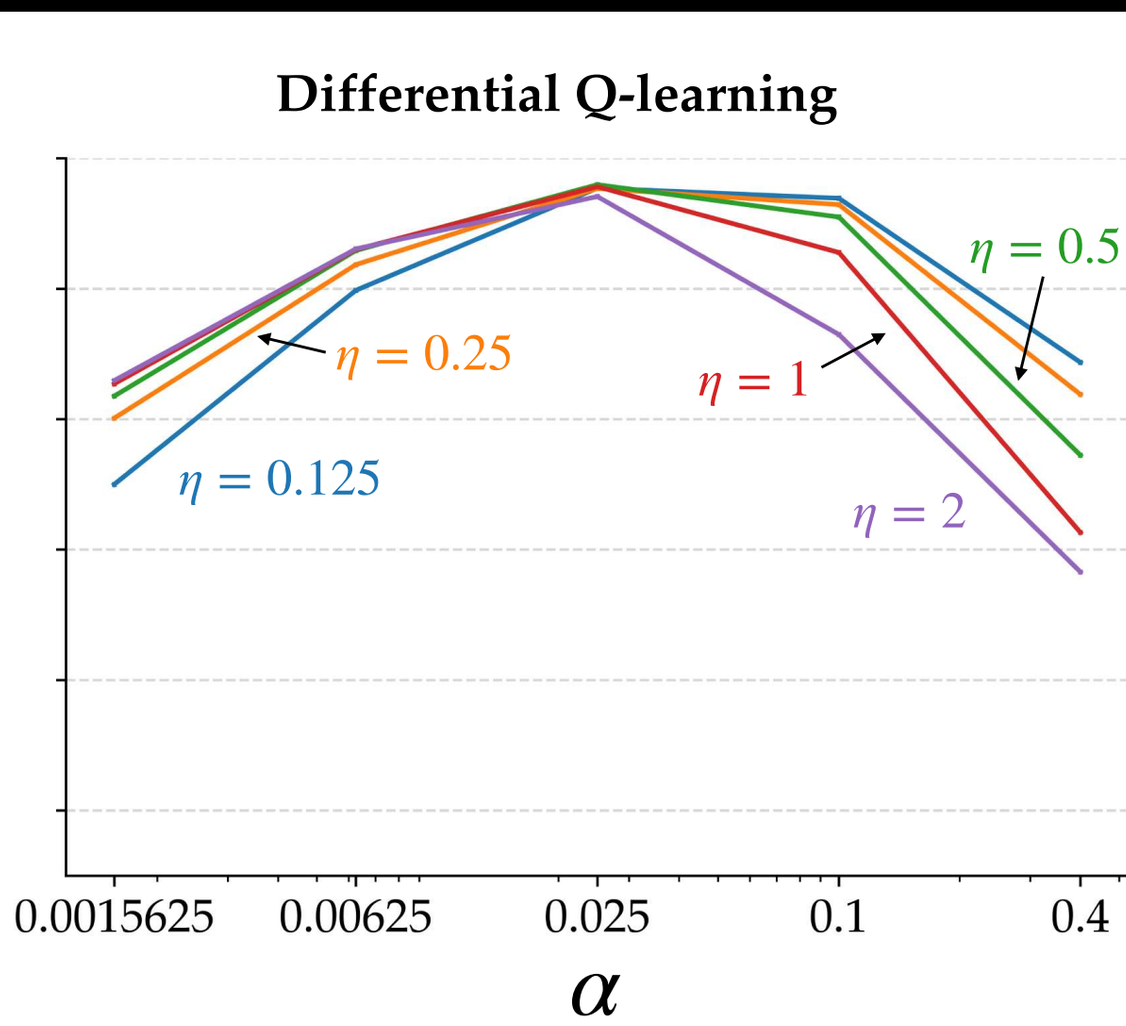


EXPERIMENT

CONTROL

Sensitivity analysis

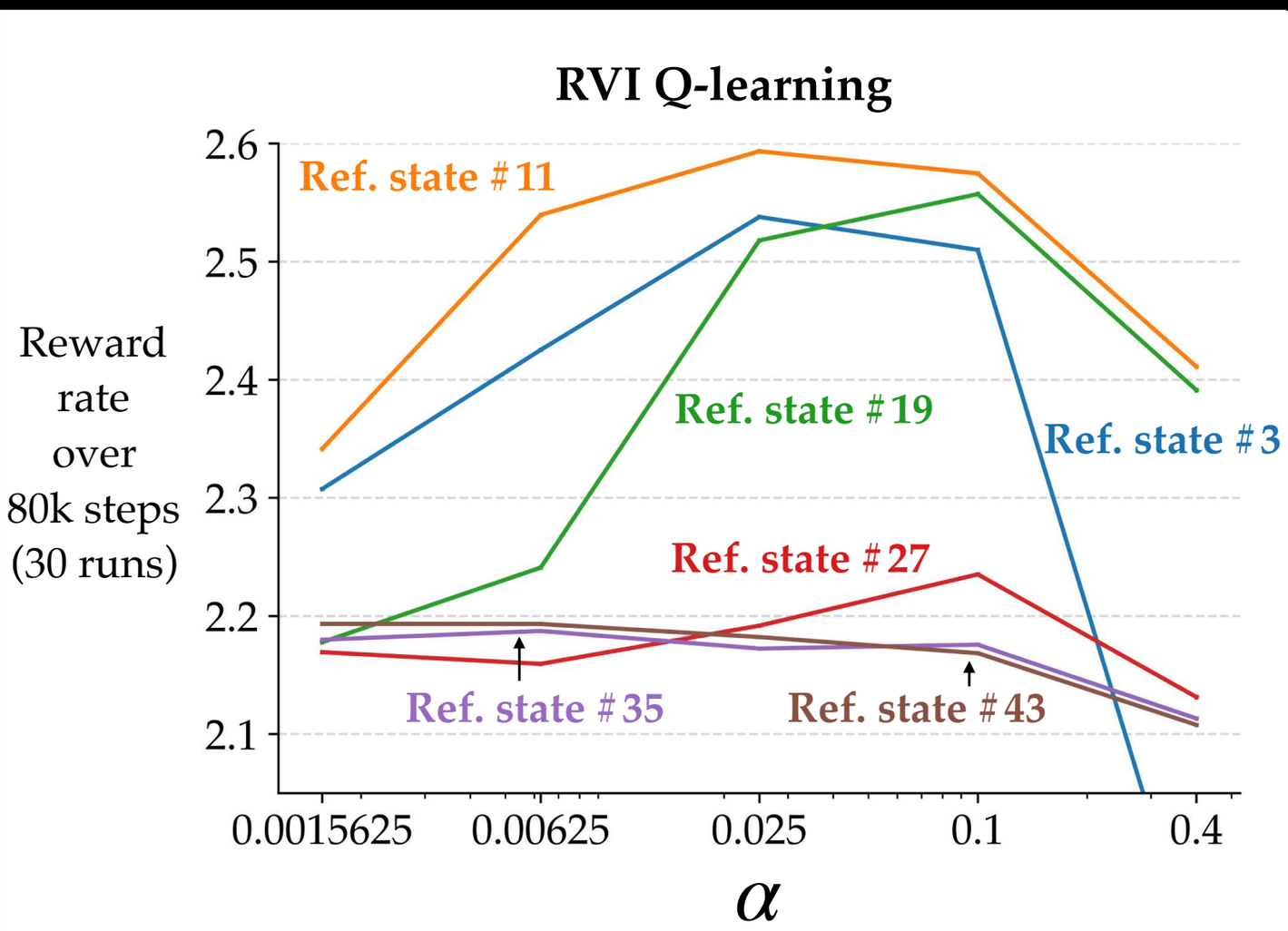


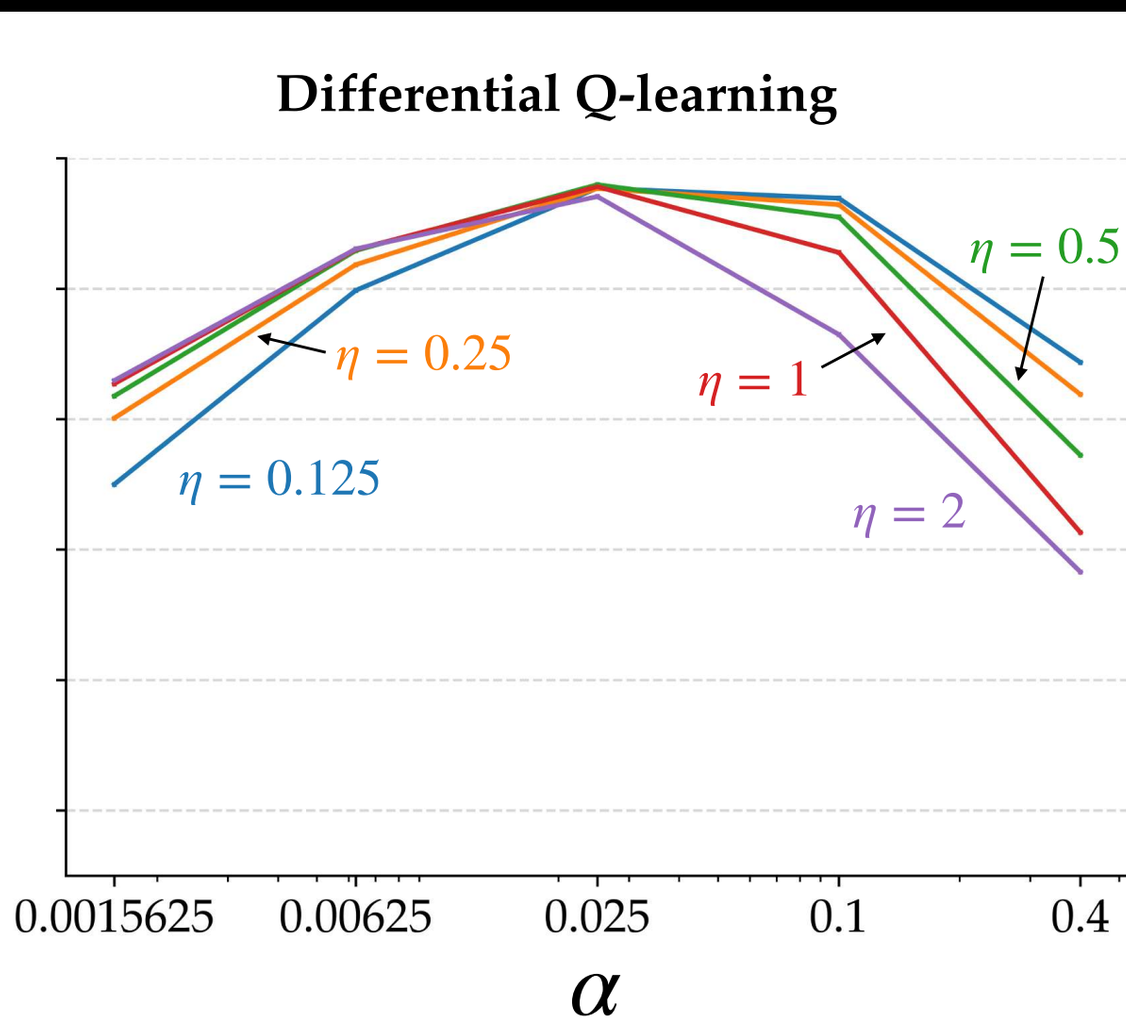


EXPERIMENT

CONTROL

Sensitivity analysis





- RVI Q-learning’s performance depends significantly on the choice of the reference state.
- Differential Q-learning’s performance varies only slightly over a wide range of parameter values.

ALGORITHM

PREDICTION

ALGORITHM

PREDICTION

$$\delta_t \doteq R_{t+1} - \bar{R}_t + V_t(S_{t+1}) - V_t(S_t)$$

ALGORITHM

PREDICTION

$$\delta_t \doteq R_{t+1} - \bar{R}_t + V_t(S_{t+1}) - V_t(S_t)$$

$$V_{t+1}(S_t) \doteq V_t(S_t) + \alpha_t \rho_t \delta_t$$

$$\bar{R}_{t+1} \doteq \bar{R}_t + \eta \alpha_t \rho_t \delta_t$$

ALGORITHM

PREDICTION

Differential TD-learning

$$\delta_t \doteq R_{t+1} - \bar{R}_t + V_t(S_{t+1}) - V_t(S_t)$$

$$V_{t+1}(S_t) \doteq V_t(S_t) + \alpha_t \rho_t \delta_t$$

$$\bar{R}_{t+1} \doteq \bar{R}_t + \eta \alpha_t \rho_t \delta_t$$

ALGORITHM

PREDICTION

Differential TD-learning

$$\delta_t \doteq R_{t+1} - \bar{R}_t + V_t(S_{t+1}) - V_t(S_t)$$

$$V_{t+1}(S_t) \doteq V_t(S_t) + \alpha_t \rho_t \delta_t$$

$$\bar{R}_{t+1} \doteq \bar{R}_t + \eta \alpha_t \rho_t \delta_t$$

Theorem 2 (sketch)

If 1) the MDP is unichain,
2) the stepsizes are decreased appropriately,
3) all the states are updated infinite number of times,
4) the maximum ratio of the update frequencies is finite,
5) b covers all the actions that π may choose in all states,

then the Differential TD-learning algorithm converges a.s.:
$\bar{R}_t$ to $r(\pi)$, V_t to a solution of the Bellman equation.

ALGORITHM

PREDICTION

ALGORITHM

PREDICTION

Algorithm 3: Differential TD-learning (one-step off-policy prediction)

Input: The policy π to be evaluated, and b to be used
Algorithm parameters: step sizes α, η
Initialize $V(s)\ \forall s,\ \bar{R}$ arbitrarily (e.g., to zero)
while *still time to train* **do**
    $A \leftarrow$ action given by b for S
    Take action A, observe R, S'
    $\delta = R - \bar{R} + V(S') - V(S)$
    $\rho = \frac{\pi(A|S)}{b(A|S)}$
    $V(S) = V(S) + \alpha\rho\delta$
    $\bar{R} = \bar{R} + \eta\alpha\rho\delta$
    $S = S'$
end
return V

ALGORITHM

PREDICTION

Algorithm 3: Differential TD-learning (one-step off-policy prediction)

Input: The policy π to be evaluated, and b to be used
Algorithm parameters: step sizes α, η
Initialize $V(s)\ \forall s,\ \bar{R}$ arbitrarily (e.g., to zero)
while *still time to train* **do**
  $A \leftarrow$ action given by b for S
  Take action A, observe R, S'
  $\delta = R - \bar{R} + V(S') - V(S)$
  $\rho = \frac{\pi(A|S)}{b(A|S)}$
  $V(S) = V(S) + \alpha\rho\delta$
  $\bar{R} = \bar{R} + \eta\alpha\rho\delta$
  $S = S'$
end
return V

Average Cost TD-learning

$$\bar{R}_{t+1} \doteq \bar{R}_t + \eta\alpha_t(R_{t+1} - \bar{R}_t)$$

PREDICTION

Algorithm 3: Differential TD-learning (one-step off-policy prediction)

Input: The policy π to be evaluated, and b to be used
Algorithm parameters: step sizes α, η

Initialize $V(s)\ \forall s,\ \bar{R}$ arbitrarily (e.g., to zero)
while *still time to train* **do**
  $A \leftarrow$ action given by b for S
  Take action A, observe R, S'
  $\delta = R - \bar{R} + V(S') - V(S)$
  $\rho = \frac{\pi(A|S)}{b(A|S)}$
  $V(S) = V(S) + \alpha\rho\delta$
  $\bar{R} = \bar{R} + \eta\alpha\rho\delta$
  $S = S'$
end
return V

Average Cost TD-learning

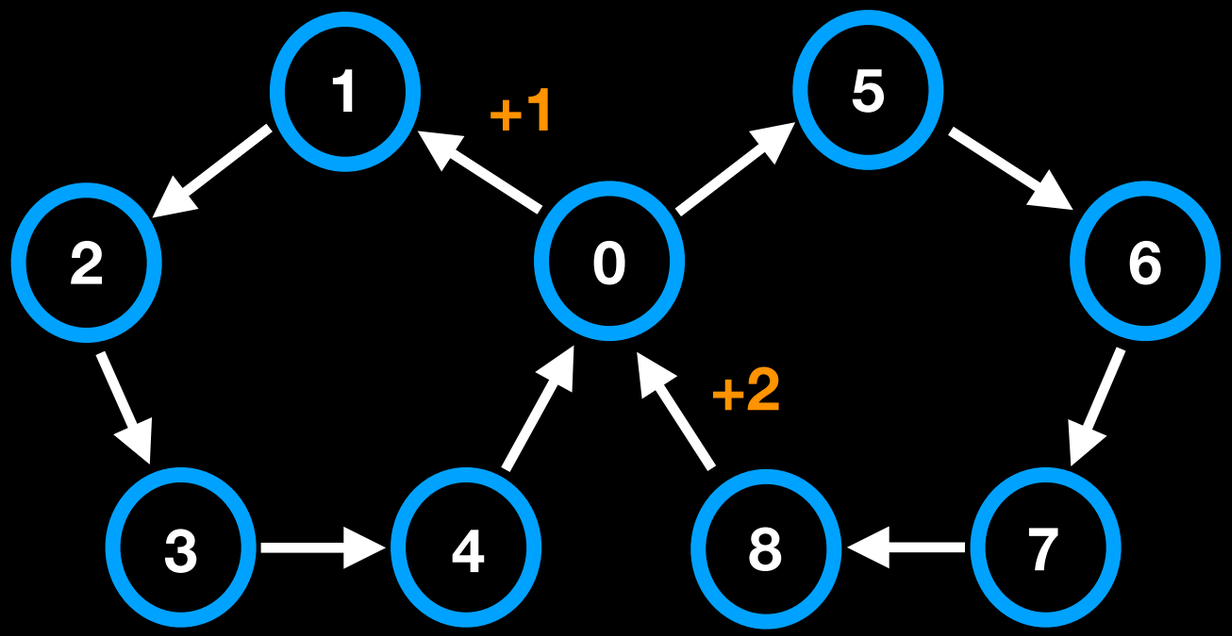
$$\bar{R}_{t+1} \doteq \bar{R}_t + \eta\alpha_t(R_{t+1} - \bar{R}_t)$$

(restricted to on-policy)

ENVIRONMENT

PREDICTION

- Two Loop Task



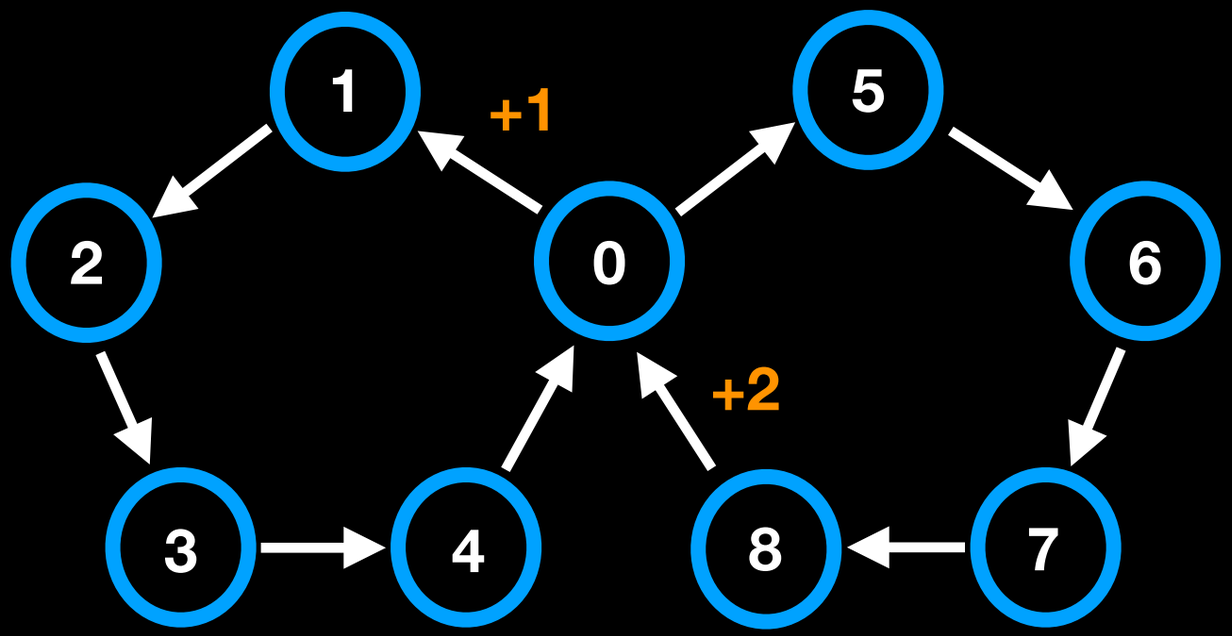
EXPERIMENT

- $\pi_0 = [0.5,\ 0.5],\ b_0 = [0.9,\ 0.1]$
- $\alpha \in \{0.025,\ 0.05,\ 0.1,\ 0.2,\ 0.4\}$
- $\eta \in \{0.125,\ 0.25,\ 0.5,\ 1,\ 2\}$
- $\epsilon = 0.1$
- 10,000 steps
- 30 runs

ENVIRONMENT

PREDICTION

- Two Loop Task



EXPERIMENT

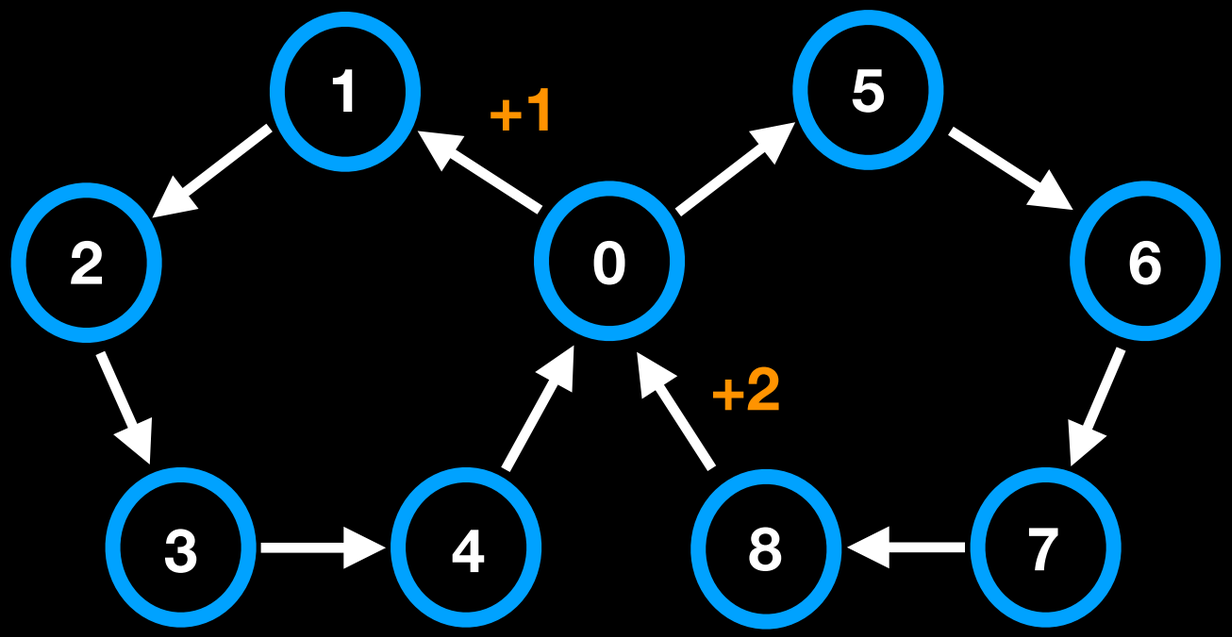
- $\pi_0 = [0.5,\ 0.5],\ b_0 = [0.9,\ 0.1]$
- $\alpha \in \{0.025,\ 0.05,\ 0.1,\ 0.2,\ 0.4\}$
- $\eta \in \{0.125,\ 0.25,\ 0.5,\ 1,\ 2\}$
- $\epsilon = 0.1$
- 10,000 steps
- 30 runs

- Evaluation metric:
  - RMSVE

$$\| v - \quad v_\pi \qquad \|_{d_\pi}$$

PREDICTION

- Two Loop Task



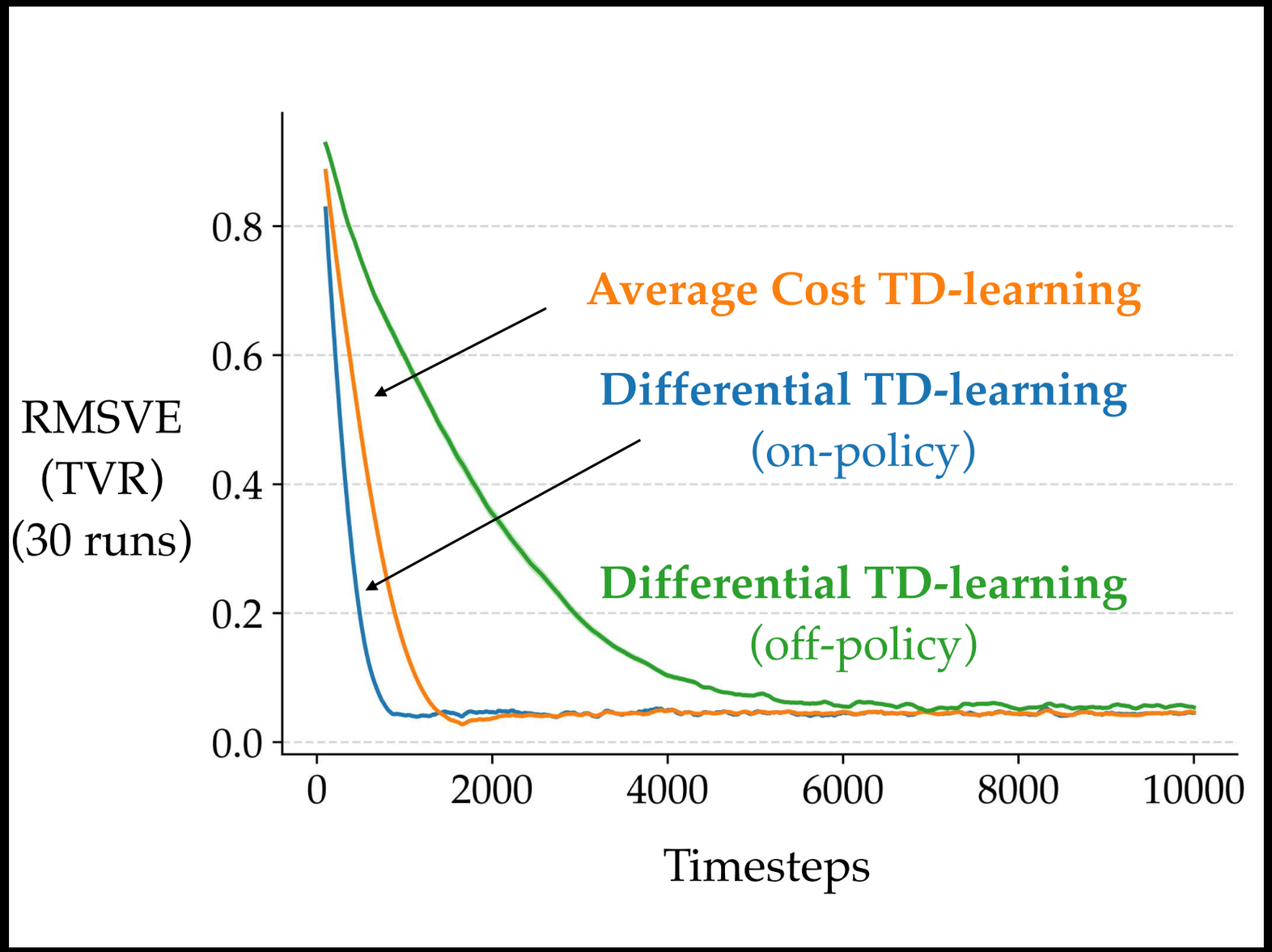
- $\pi_0 = [0.5, 0.5]$, $b_0 = [0.9, 0.1]$
- $\alpha \in \{0.025, 0.05, 0.1, 0.2, 0.4\}$
- $\eta \in \{0.125, 0.25, 0.5, 1, 2\}$
- $\epsilon = 0.1$
- 10,000 steps
- 30 runs

- Evaluation metric:
  - RMSVE

$$\inf_c \|v - (v_\pi + ce)\|_{d_\pi}$$

(Tsitsiklis and Van Roy, 1999)

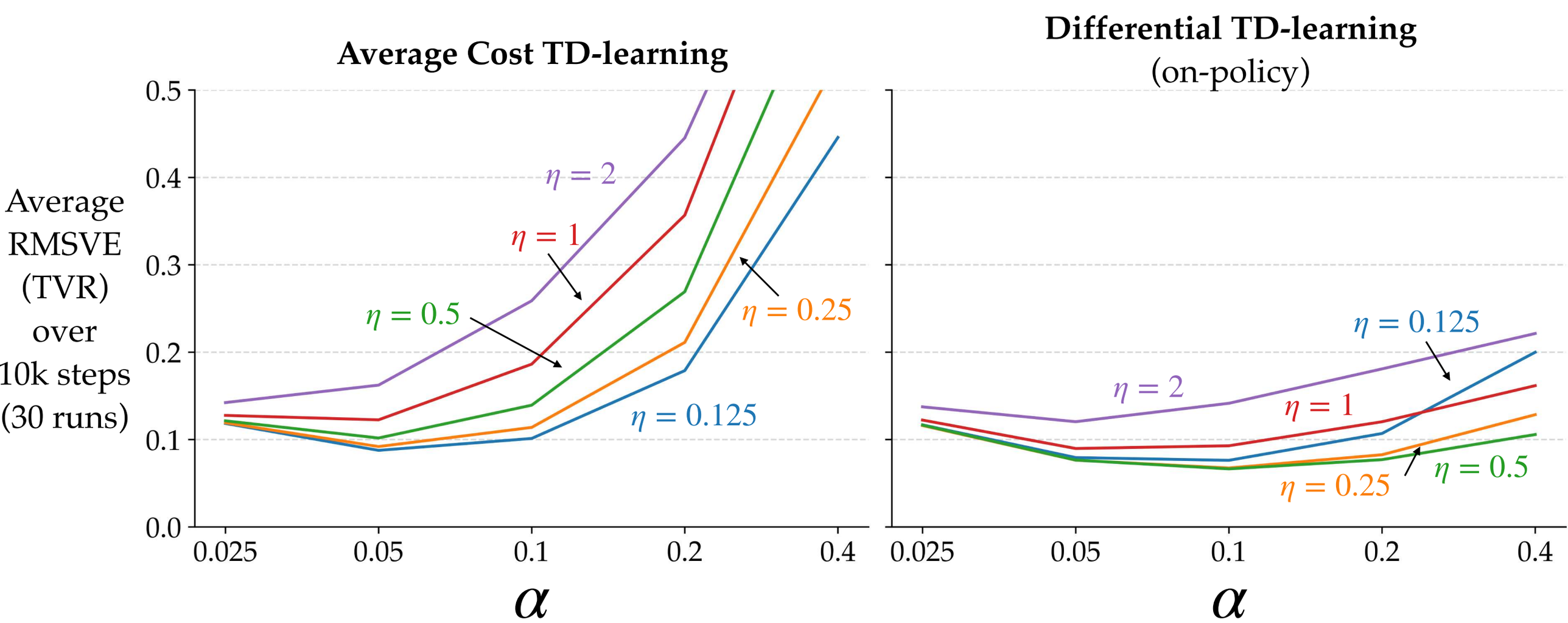
PREDICTION



Learning curves

PREDICTION

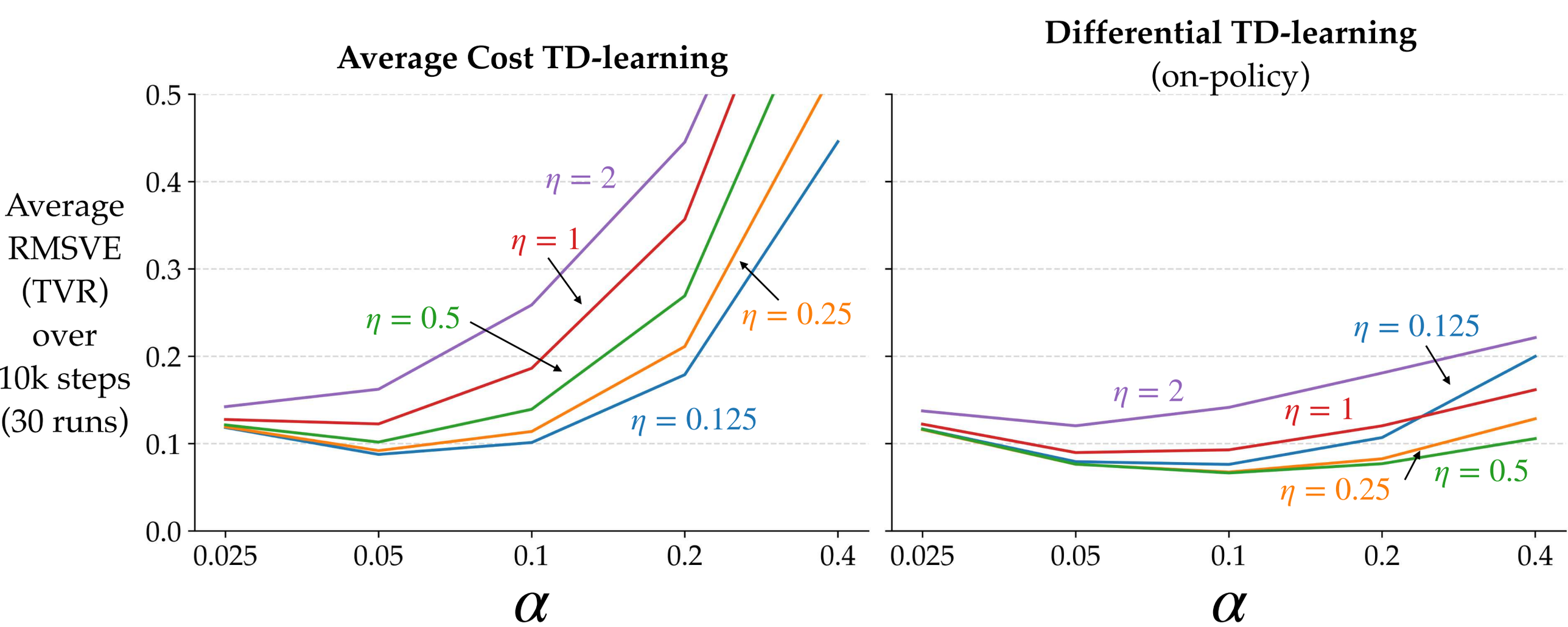
Sensitivity analysis



RESULTS

PREDICTION

Sensitivity analysis

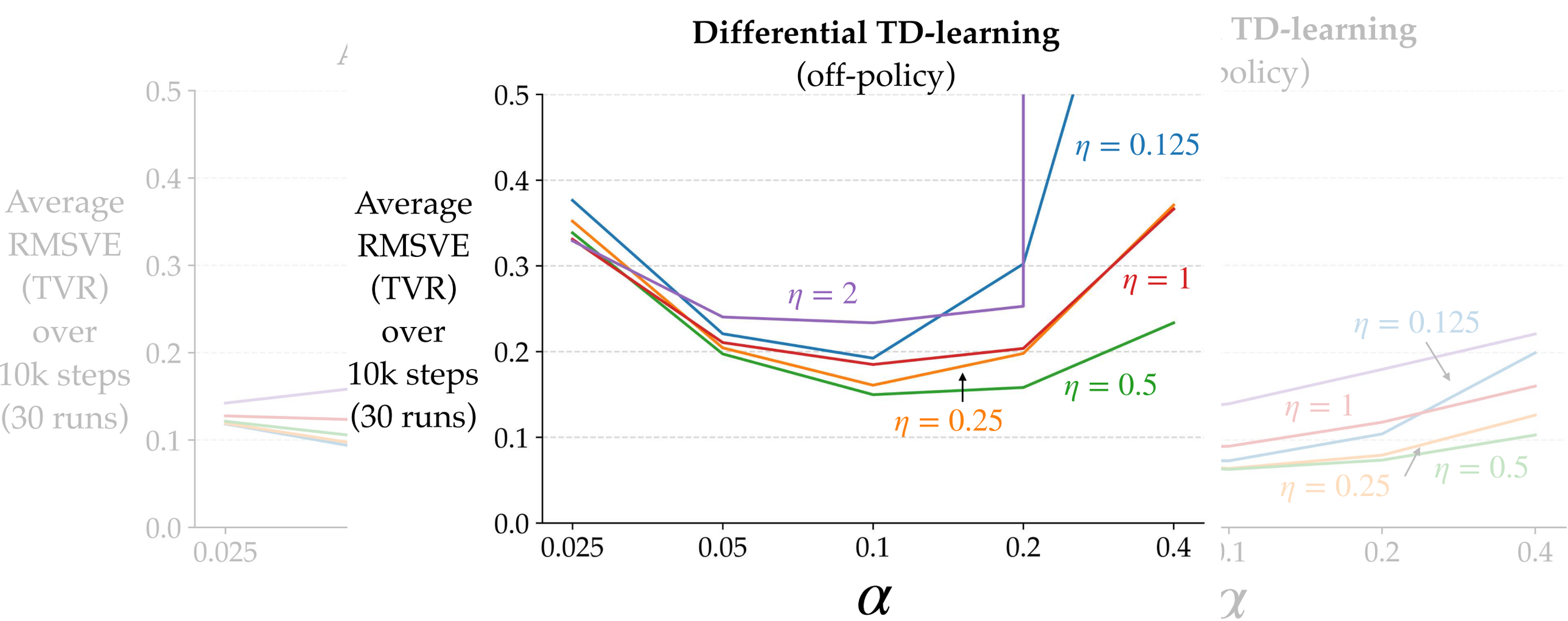


- Differential TD-learning converges faster for a wide range of parameters.

RESULTS

PREDICTION

Sensitivity analysis

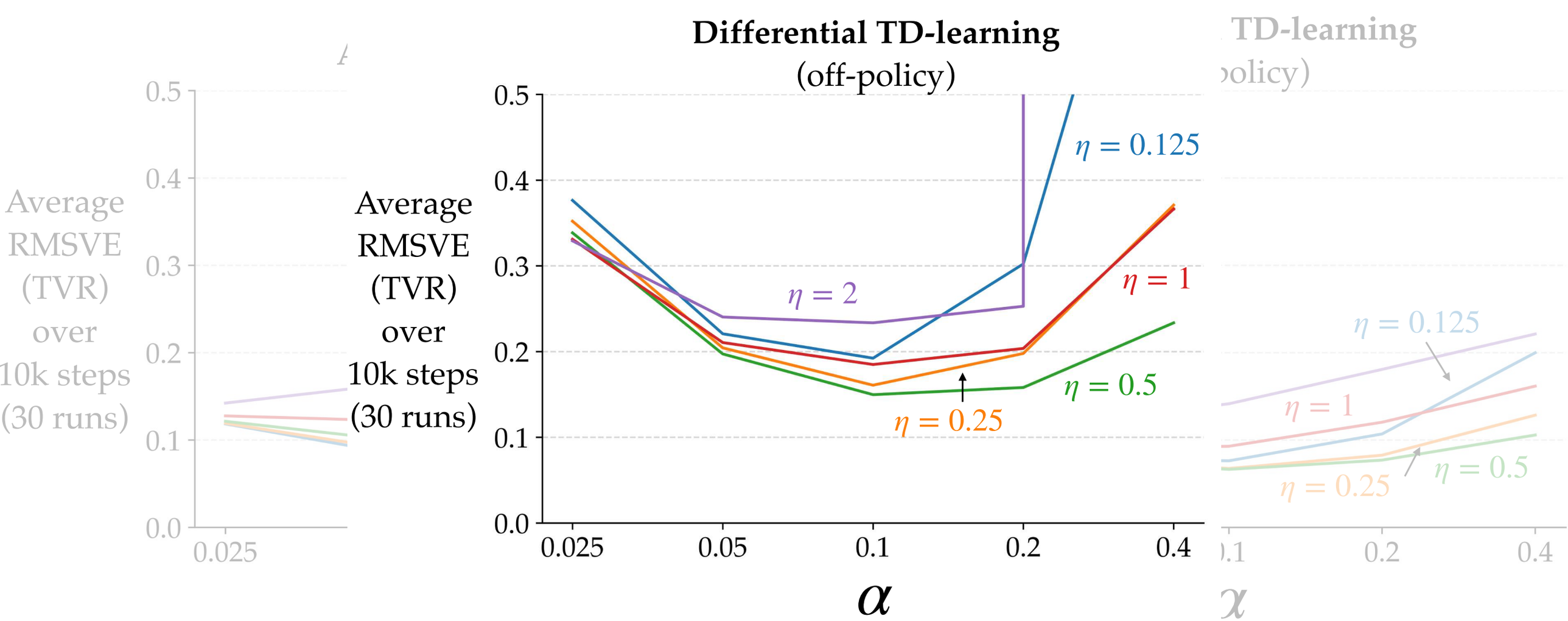


- Differential TD-learning converges faster for a wide range of parameters.

RESULTS

PREDICTION

Sensitivity analysis



- Differential TD-learning converges faster for a wide range of parameters.
- Differential TD-learning works in the off-policy setting as well.

CENTERING

CENTERING

Recall: $$v(s) = \sum_{a} \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a) \left[R_{t+1} - \bar{r} + v(s') \right] \quad \forall s$$

CENTERING

Recall: $$v(s) = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a) \left[R_{t+1} - \bar{r} + v(s') \right] \quad \forall s$$

Solutions: $$v = v_\pi + ce$$

CENTERING

Recall: $$v(s) = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a) \left[R_{t+1} - \bar{r} + v(s') \right] \quad \forall s$$

Solutions: $$v = v_\pi + ce$$

Lemma

$$d_\pi^T v_\pi = 0,$$

i.e., the average of the differential value function is zero.

CENTERING

Recall: $$v(s) = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a) \left[R_{t+1} - \bar{r} + v(s') \right] \quad \forall s$$

Solutions: $$v = v_\pi + ce$$

Lemma

$$d_\pi^T v_\pi = 0,$$

i.e., the average of the differential value function is zero.

$\Longrightarrow$ there is only one *centered* differential value function

CENTERING

Recall: $$v(s) = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a) \left[R_{t+1} - \bar{r} + v(s') \right] \quad \forall s$$

Solutions: $$v = v_\pi + ce$$

Lemma

$$d_\pi^T v_\pi = 0,$$

i.e., the average of the differential value function is zero.

$\Longrightarrow$ there is only one *centered* differential value function $v = v_\pi + ce$

CENTERING

Recall: $v(s) = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a) \left[R_{t+1} - \bar{r} + v(s') \right] \quad \forall s$

Solutions: $v = v_\pi + ce$

Lemma

$$d_\pi^T v_\pi = 0,$$

i.e., the average of the differential value function is zero.

$\Longrightarrow$ there is only one *centered* differential value function

$$v = v_\pi + ce$$
$$\Longrightarrow \quad c = d_\pi^T v$$

CENTERING

Recall: $$v(s) = \sum_a \pi(a \mid s) \sum_{s',r} p(s', r \mid s, a) \left[R_{t+1} - \bar{r} + v(s') \right] \quad \forall s$$

Solutions: $$v = v_\pi + ce$$

Lemma

$$d_\pi^T v_\pi = 0,$$

i.e., the average of the differential value function is zero.

$\Longrightarrow$ there is only one *centered* differential value function

$$v = v_\pi + ce$$
$$\Longrightarrow \quad c = d_\pi^T v$$
$$r(\pi) = d_\pi^T r_\pi$$

ALGORITHM

CENTERING

$$\delta_t \doteq R_{t+1} - \bar{R}_t + V_t(S_{t+1}) - V_t(S_t)$$

$$V_{t+1}(S_t) \doteq V_t(S_t) + \alpha_t \rho_t \delta_t$$

$$\bar{R}_{t+1} \doteq \bar{R}_t + \eta \alpha_t \rho_t \delta_t$$

System 1

ALGORITHM

CENTERING

$$\delta_t \doteq R_{t+1} - \bar{R}_t + V_t(S_{t+1}) - V_t(S_t)$$

$$V_{t+1}(S_t) \doteq V_t(S_t) + \alpha_t \rho_t \delta_t$$

$$\bar{R}_{t+1} \doteq \bar{R}_t + \eta \alpha_t \rho_t \delta_t$$

System 1

$$\Delta_t \doteq V_t(S_t) - \bar{V}_t + F_t(S_{t+1}) - F_t(S_t)$$

$$F_{t+1}(S_t) \doteq F_t(S_t) + \beta_t \rho_t \Delta_t$$

$$\bar{V}_{t+1} \doteq \bar{V}_t + \kappa \beta_t \rho_t \Delta_t$$

System 2

CENTERING

$$\delta_t \doteq R_{t+1} - \bar{R}_t + V_t(S_{t+1}) - V_t(S_t)$$

$$V_{t+1}(S_t) \doteq V_t(S_t) + \alpha_t \rho_t \delta_t$$

$$\bar{R}_{t+1} \doteq \bar{R}_t + \eta \alpha_t \rho_t \delta_t$$

System 1

$$\Delta_t \doteq V_t(S_t) - \bar{V}_t + F_t(S_{t+1}) - F_t(S_t)$$

$$F_{t+1}(S_t) \doteq F_t(S_t) + \beta_t \rho_t \Delta_t$$

$$\bar{V}_{t+1} \doteq \bar{V}_t + \kappa \beta_t \rho_t \Delta_t$$

System 2

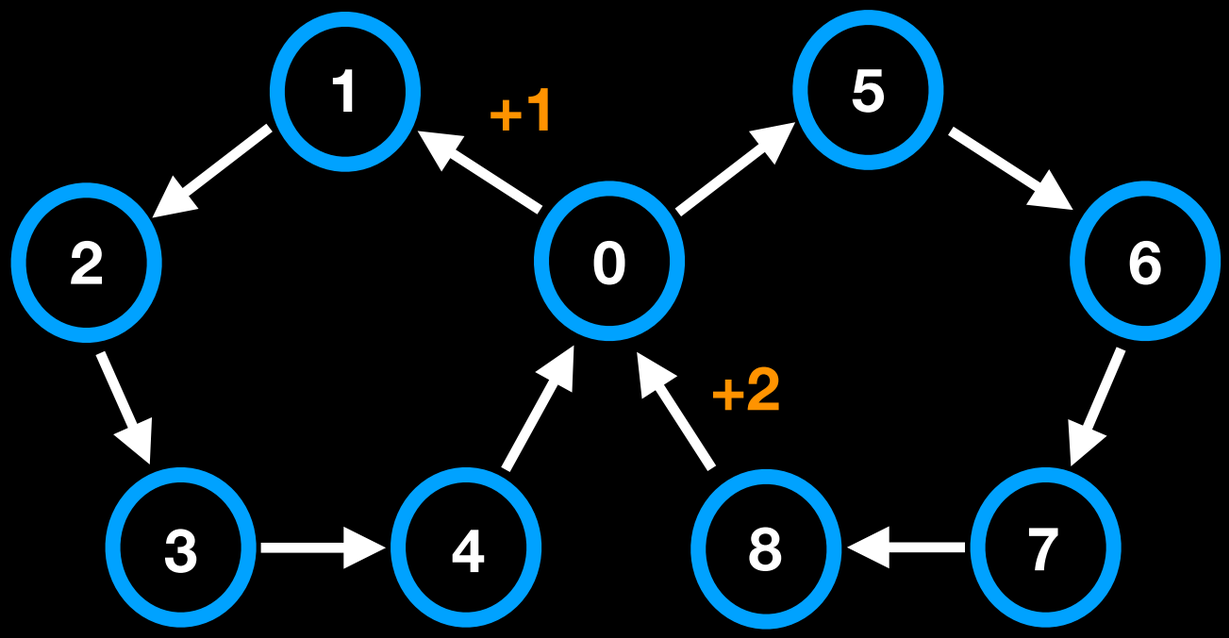
Theorem 3 (sketch)

If the previous assumptions hold, then the Centered Differential TD-learning algorithm converges a.s.:
$\bar{R}_t$ to $r(\pi)$, $V_t - \bar{V}_t\, e$ to the centered differential value function

ENVIRONMENT

CENTERING

- Two Loop Task



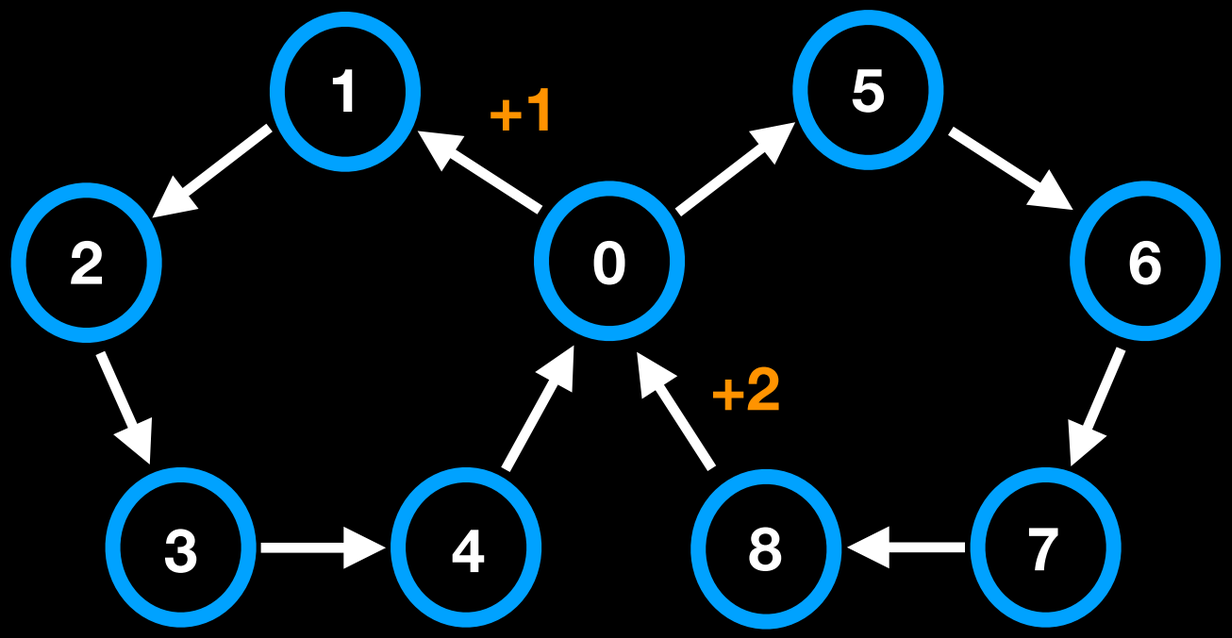
EXPERIMENT

- $\beta \in \{0.025,\ 0.05,\ 0.1,\ 0.2,\ 0.4\}$
- $\kappa \in \{0.125,\ 0.25,\ 0.5,\ 1,\ 2\}$
- $\epsilon = 0.1$
- 10,000 steps
- 30 runs

ENVIRONMENT

CENTERING

- Two Loop Task



EXPERIMENT

- $\beta \in \{0.025, 0.05, 0.1, 0.2, 0.4\}$
- $\kappa \in \{0.125, 0.25, 0.5, 1, 2\}$
- $\epsilon = 0.1$
- 10,000 steps
- 30 runs

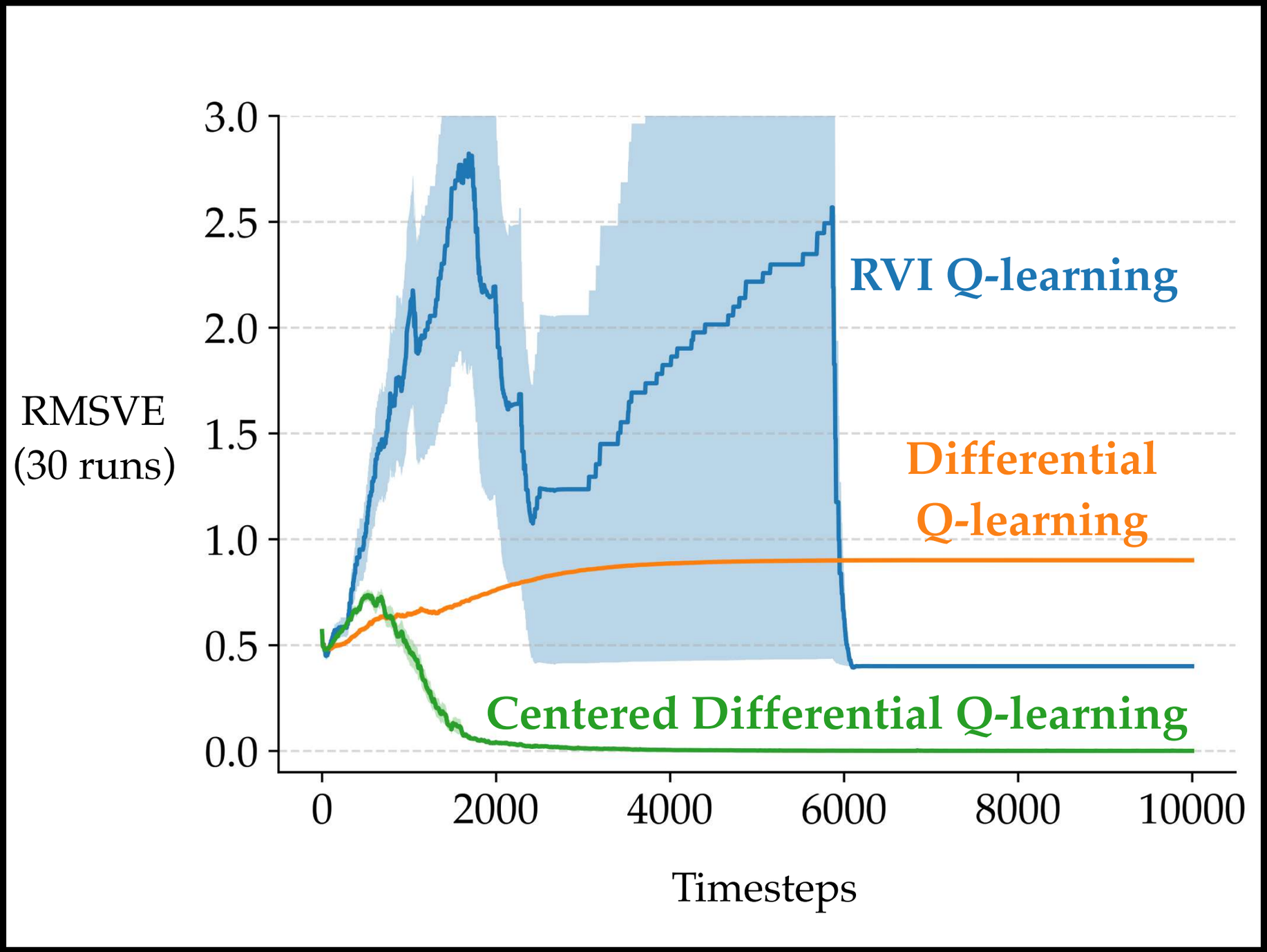
- Evaluation metric:
  - RMSVE

$$\|v - v_\pi\|_{d_\pi}$$

(the usual one)

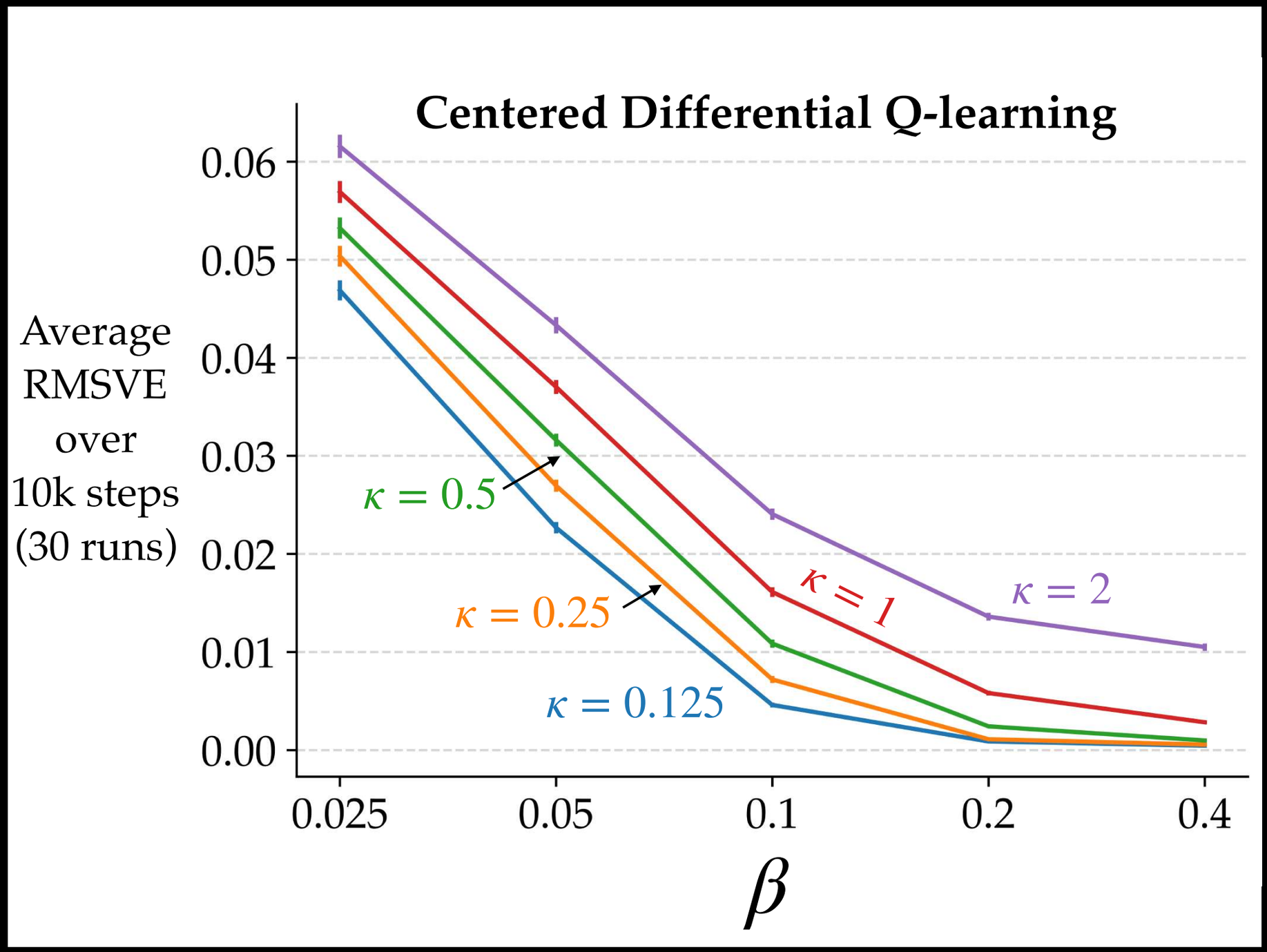
RESULTS

CENTERING



Learning curves

CENTERING



Sensitivity analysis

CONCLUSIONS

CONCLUSIONS

- The first general proven-convergent off-policy model-free *control* algorithm without reference states

CONCLUSIONS

- The first general proven-convergent off-policy model-free *control* algorithm without reference states
- The first proven-convergent off-policy model-free *prediction* algorithm

CONCLUSIONS

- The first general proven-convergent off-policy model-free *control* algorithm without reference states
- The first proven-convergent off-policy model-free *prediction* algorithm
- A general technique to estimate the actual *centered* value function rather than the value function plus an offset

CONCLUSIONS

- The first general proven-convergent off-policy model-free *control* algorithm without reference states
- The first proven-convergent off-policy model-free *prediction* algorithm
- A general technique to estimate the actual *centered* value function rather than the value function plus an offset
  - All of our learning algorithms are fully online, and all of our planning algorithms are fully incremental

CONCLUSIONS

- The first general proven-convergent off-policy model-free *control* algorithm without reference states
- The first proven-convergent off-policy model-free *prediction* algorithm
- A general technique to estimate the actual *centered* value function rather than the value function plus an offset
  - All of our learning algorithms are fully online, and all of our planning algorithms are fully incremental
  - Empirically, the use of the temporal-difference error generally results in faster learning in the domains tested, and reliance on a reference state generally results in slower learning and risks divergence.

FUTURE WORK

FUTURE WORK

- Extension of these tabular algorithms to function approximation

FUTURE WORK

- Extension of these tabular algorithms to function approximation
  - currently working on linear FA, both learning and planning

FUTURE WORK

- Extension of these tabular algorithms to function approximation
  - currently working on linear FA, both learning and planning
  - deadly triad :/

FUTURE WORK

- Extension of these tabular algorithms to function approximation
  - currently working on linear FA, both learning and planning
  - deadly triad :/
- Extension to SMDPs so they can be used with temporal abstractions like options

FUTURE WORK

- Extension of these tabular algorithms to function approximation
  - currently working on linear FA, both learning and planning
  - deadly triad :/
- Extension to SMDPs so they can be used with temporal abstractions like options
- Does centering the features stabilize (off-policy) learning?

QUESTIONS?

Paper: *https://arxiv.org/abs/2006.16318*
Code: *https://github.com/abhisheknaik96/average-reward-methods*

INTUITION: WHY USING TD ERROR MIGHT BE BETTER

INTUITION: WHY USING TD ERROR MIGHT BE BETTER

- Example 10.8 from the book