

A Walk in the Park: Learning to Walk in 20 Minutes With Model—Free Reinforcement Learning



He Zhi, CSE of SYSU

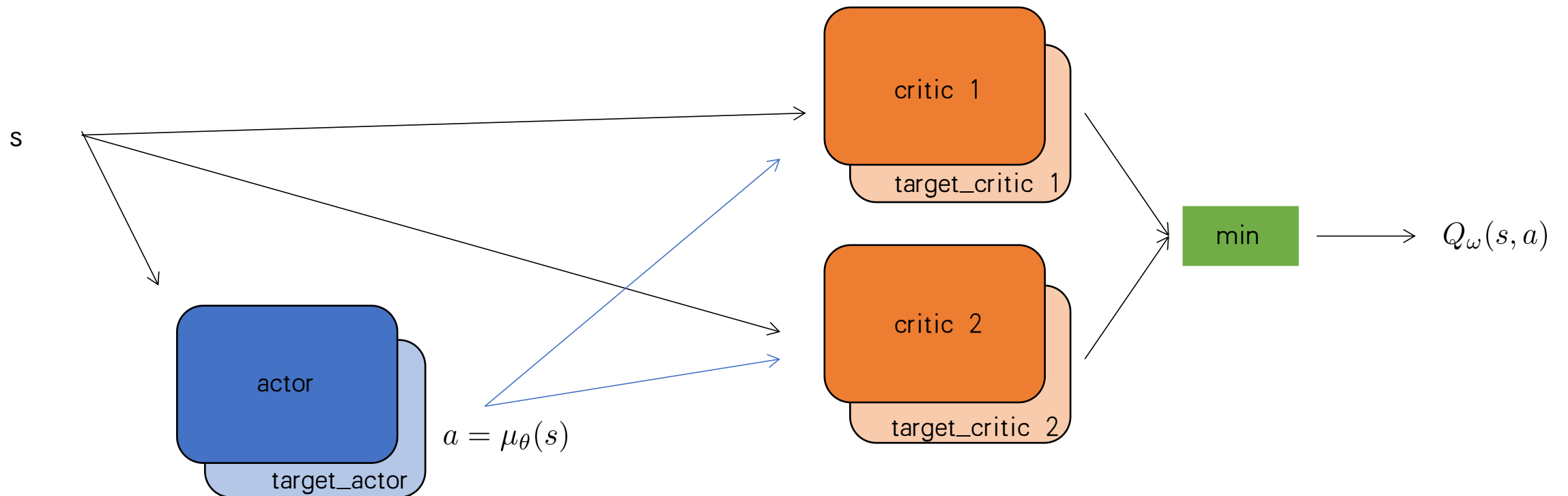
Email: hezh58@mail2.sysu.edu.cn

Learning inefficiency

Regularization/Normalization

Preliminaries: SAC (soft actor critic)

- A max-entropy DRL: $\pi = \arg \max_{\pi} \mathbb{E}[\sum_t r(s_t, a_t) + \alpha H(\pi(\Delta|s_t))]$



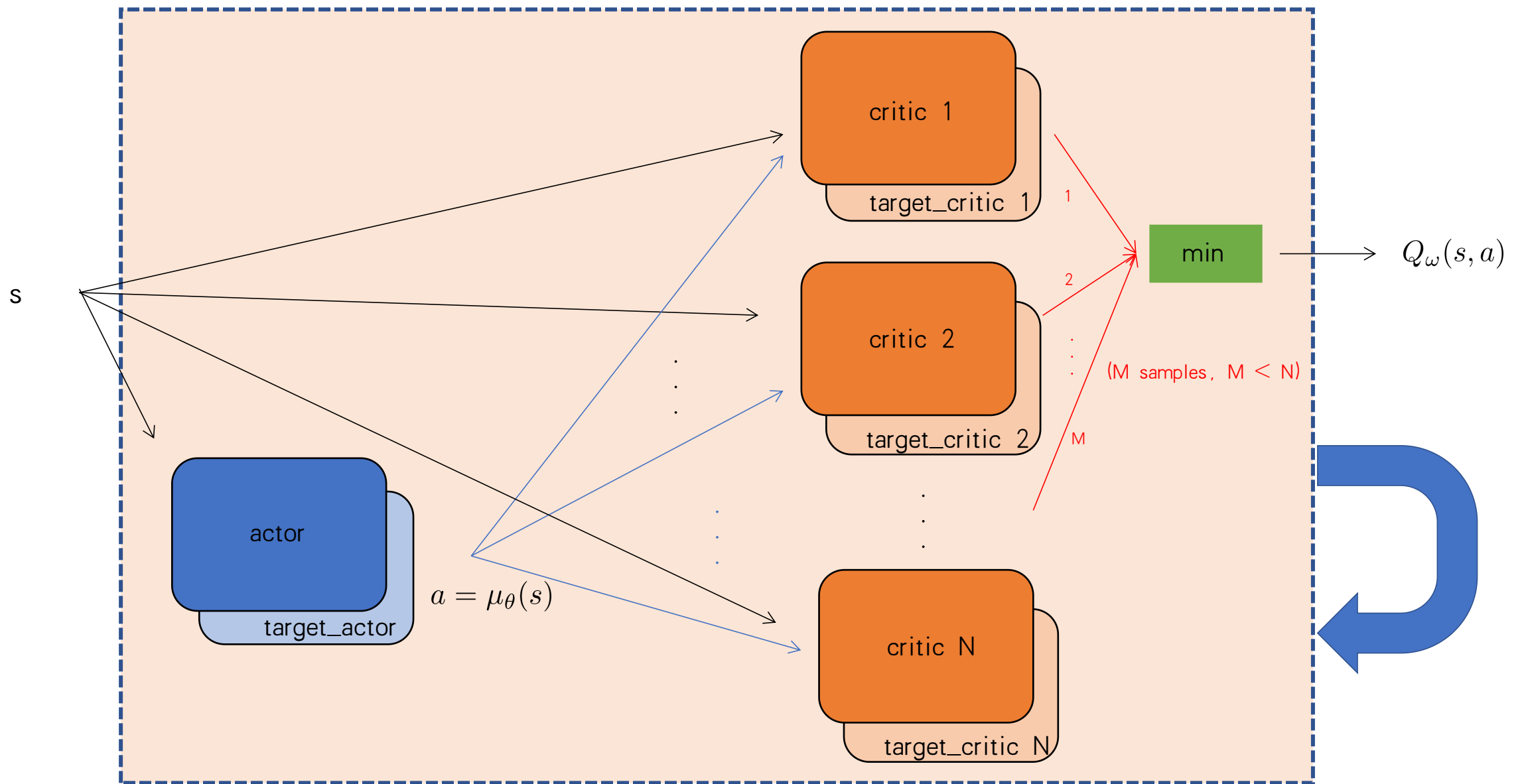
REDQ

UTD: Update-To-Data ratio

$$= \frac{\#gradient \ steps}{\#environment - interacting \ steps}$$

- N: number of ensemble of Q networks
- M: number of randomized sampling Q networks to minimise
- G: number of updating epochs

REDQ



REDQ

Algorithm 1 Randomized Ensembled Double Q-learning (REDQ)

- 1: Initialize policy parameters θ , N Q-function parameters ϕ_i , $i = 1, \dots, N$, empty replay buffer \mathcal{D} . Set target parameters $\phi_{\text{targ},i} \leftarrow \phi_i$, for $i = 1, 2, \dots, N$
- 2: **repeat**
- 3: Take one action $a_t \sim \pi_\theta(\cdot | s_t)$. Observe reward r_t , new state s_{t+1} .
- 4: Add data to buffer: $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, r_t, s_{t+1})\}$
- 5: **for** G updates **do**
- 6: Sample a mini-batch $B = \{(s, a, r, s')\}$ from \mathcal{D}
- 7: Sample a set \mathcal{M} of M distinct indices from $\{1, 2, \dots, N\}$
- 8: Compute the Q target y (same for all of the N Q-functions):

$$y = r + \gamma \left(\min_{i \in \mathcal{M}} Q_{\phi_{\text{targ},i}}(s', \tilde{a}') - \alpha \log \pi_\theta(\tilde{a}' | s') \right), \quad \tilde{a}' \sim \pi_\theta(\cdot | s')$$

- 9: **for** $i = 1, \dots, N$ **do**
- 10: Update ϕ_i with gradient descent using

$$\nabla_{\phi} \frac{1}{|B|} \sum_{(s,a,r,s') \in B} (Q_{\phi_i}(s, a) - y)^2$$

- 11: Update target networks with $\phi_{\text{targ},i} \leftarrow \rho \phi_{\text{targ},i} + (1 - \rho) \phi_i$
- 12: Update policy parameters θ with gradient ascent using

$$\nabla_{\theta} \frac{1}{|B|} \sum_{s \in B} \left(\frac{1}{N} \sum_{i=1}^N Q_{\phi_i}(s, \tilde{a}_\theta(s)) - \alpha \log \pi_\theta(\tilde{a}_\theta(s) | s) \right), \quad \tilde{a}_\theta(s) \sim \pi_\theta(\cdot | s)$$

- SAC: $G = 1, N = M = 2$

$$\text{UTD} = 1$$

- REDQ: $G > 1, N > M \gg 2$

$$\text{UTD} = 20$$

DroQ

- small ensemble
- dropout Q-network

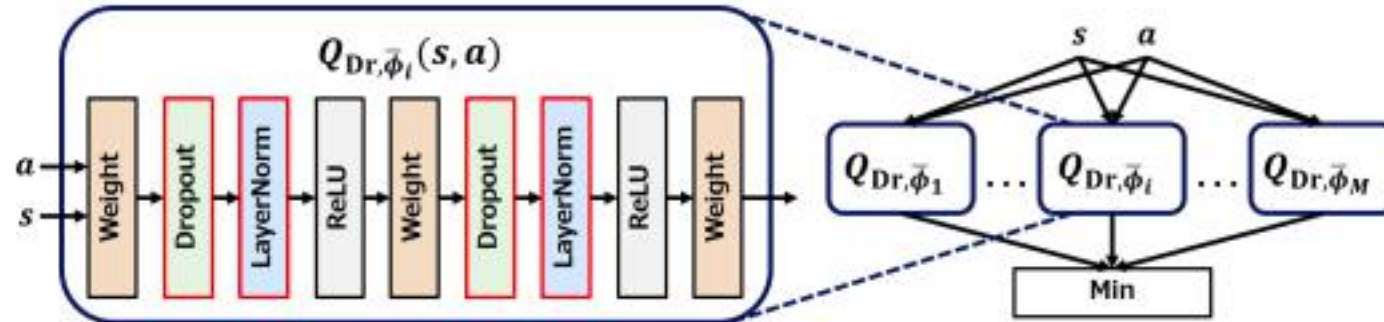


Figure 1: Dropout Q-function implementation (left part) and how dropout Q-functions are used in target (right part). **Dropout Q-function implementation:** Our dropout function is implemented by modifying that used by Chen et al. (2021b). Our modification (highlighted in red) is adding dropout (Dropout) and layer normalization (LayerNorm). “Weight” is a weight layer and “ReLU” is the activation layer of rectified linear units. Parameters $\bar{\phi}_i$ represent the weights and biases in weight layers. **How dropout Q-functions are used in target:** M dropout Q-functions are used to calculate the target value as $\min_{i=1, \dots, M} Q_{\text{Dr}, \bar{\phi}_i}(s, a)$.

DroQ

Algorithm 2 DroQ

- 1: Initialize policy parameters θ , M Q-function parameters ϕ_i , $i = 1, \dots, M$, and empty replay buffer \mathcal{D} . Set target parameters $\bar{\phi}_i \leftarrow \phi_i$, for $i = 1, \dots, M$.
- 2: **repeat**
- 3: Take action $a_t \sim \pi_\theta(\cdot|s_t)$. Observe reward r_t , next state s_{t+1} ; $\mathcal{D} \leftarrow \mathcal{D} \cup (s_t, a_t, r_t, s_{t+1})$.
- 4: **for** G updates **do**
- 5: Sample a mini-batch $\mathcal{B} = \{(s, a, r, s')\}$ from \mathcal{D} .
- 6: Compute the Q target y for the dropout Q-functions:

$$y = r + \gamma \left(\min_{i=1, \dots, M} Q_{\text{Dr}, \bar{\phi}_i}(s', a') - \alpha \log \pi_\theta(a'|s') \right), \quad a' \sim \pi_\theta(\cdot|s')$$

- 7: **for** $i = 1, \dots, M$ **do**
- 8: Update ϕ_i with gradient descent using

$$\nabla_{\phi} \frac{1}{|\mathcal{B}|} \sum_{(s, a, r, s') \in \mathcal{B}} (Q_{\text{Dr}, \phi_i}(s, a) - y)^2$$

- 9: Update target networks with $\bar{\phi}_i \leftarrow \rho \bar{\phi}_i + (1 - \rho) \phi_i$.
- 10: Update θ with gradient ascent using

$$\nabla_{\theta} \frac{1}{|\mathcal{B}|} \sum_{s \in \mathcal{B}} \left(\frac{1}{M} \sum_{i=1}^M Q_{\text{Dr}, \phi_i}(s, a) - \alpha \log \pi_\theta(a|s) \right), \quad a \sim \pi_\theta(\cdot|s)$$

Experiment



Fig. 4: Examples of learned gaits acquired on a variety of real-world terrains. Left to right: flat, solid ground covered in dense foam mats; a 5cm memory foam mattress; loose ground comprised of eucalyptus bark; a grassy lawn; a gently sloped hiking trail.

- SACs
- REDQ
- DroQ

Experiment

- **state:** root orientation,
root angular velocity,
root linear velocity,
joint angles,
joint velocities,
binary foot contacts,
previous action,
...
- **action:** motor angles of joints for every leg
- **reward function:** $r(s, a) = r_v(s, a) - 0.1v_{yaw}^2$

$$r_v(s, a) \begin{cases} 1 & \text{if } v_x \in [v_t, 2v_t] \\ 0 & \text{if } v_x \in (-\infty, -v_t] \cup [4v_t, \infty] \\ 1 - \frac{|v_x - v_t|}{2v_t} & \text{otherwise.} \end{cases}$$

v_{yaw} : angular yaw velocity

v_t : target velocity

Result

(c) SAC variants

