Soft Actor-Critic
Algorithms and Applications

Presenter: Zizhao Wang

10/12/2021

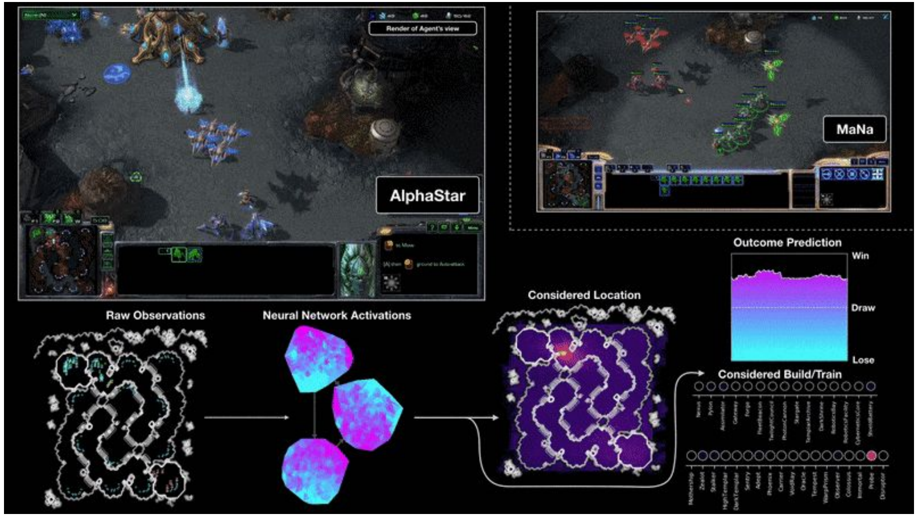
Motivation

Sample Efficiency

Reinforcement Learning (RL) can solve complex decision making and control problems, but it is notoriously sample-inefficient.

Vinyals et al., 2019

"During training, each agent experienced up to **200 years** of real-time StarCraft play."



[Grandmaster level in StarCraft II using multi-agent reinforcement learning](#)

Motivation

Sample Efficiency

Reinforcement Learning (RL) can solve complex decision making and control problems, but it is notoriously sample-inefficient.

Levine et al., 2016

- **14** robot arms learning to grasp in parallel
- Observing over 800,000 grasp attempts (**3000 robot-hours** of practice), we can see the beginnings of intelligent reactive behaviors.

[Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection]

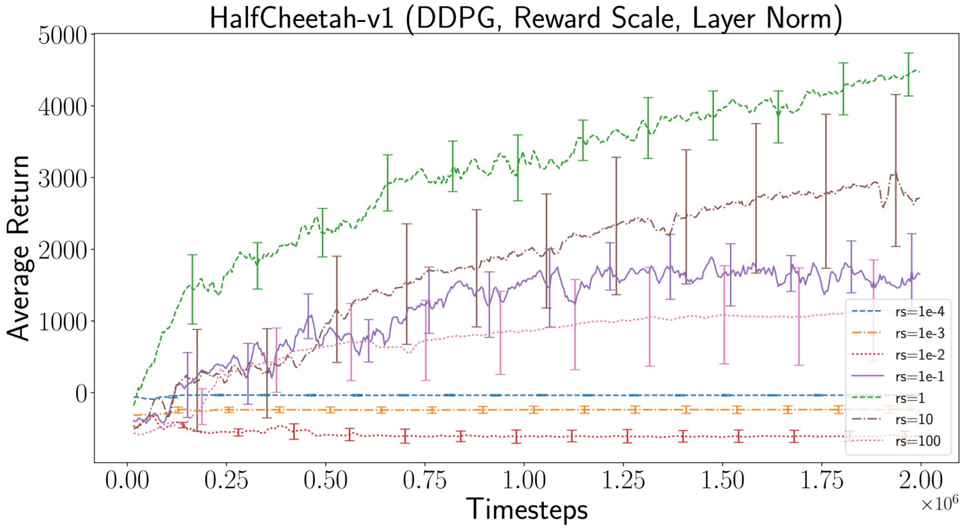
Motivation

Robustness to Hyperparameters

RL performance is brittle to hyperparameters, which needs laborious tuning.

Henderson et al., 2017

Reward scaling has a significant impact on DDPG performance.
(DDPG will be introduced soon)

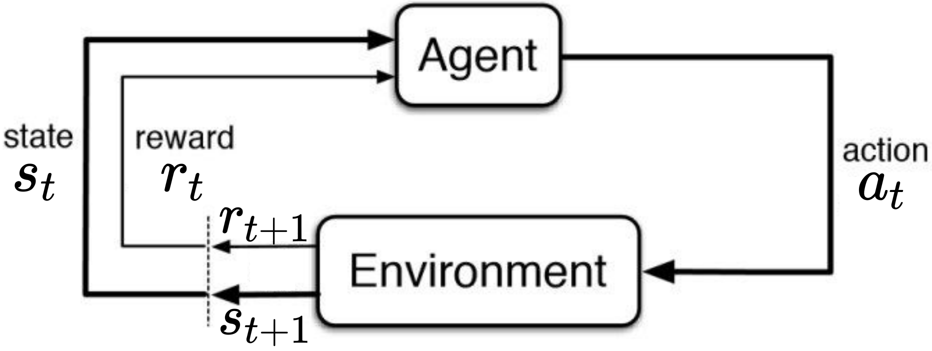


[Deep Reinforcement Learning that Matters]

Problem Setting

Markov Decision Process (MDP)

State Space	$s_t \in \mathcal{S}$
Action Space	$a_t \in \mathcal{A}$
Transition Probability	$p(s_{t+1} \mid s_t, a_t)$
Reward	$r(s_t, a_t)$
Policy	$\pi(\cdot \mid s_t)$
Trajectory	τ



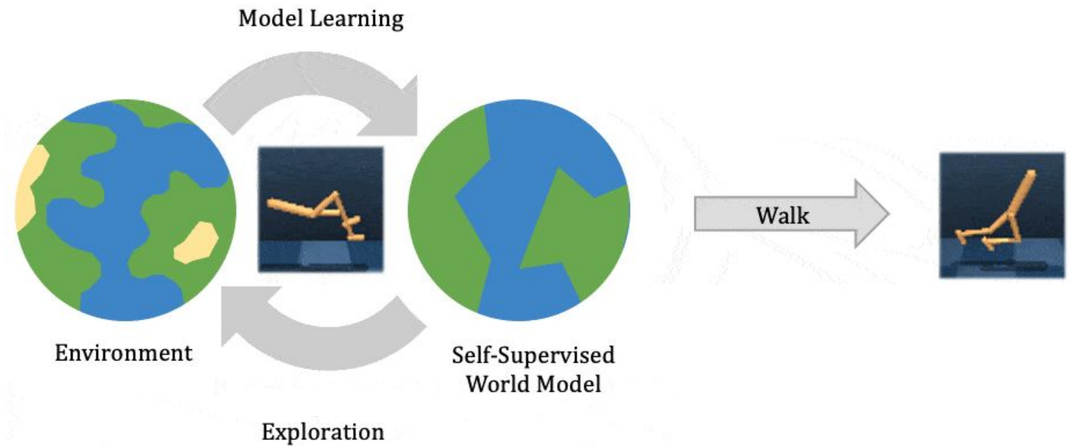
Context

RL Ranking on Sample Efficiency

better sample efficiency

- model-based

  learns a model of the environment + use the learned model to imagine interactions



Context

RL Ranking on Sample Efficiency

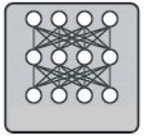
better sample efficiency

- model-based
- model-free
  - off-policy

    can learns from data generated by a

    different policy

off-policy RL



Context

RL Ranking on Sample Efficiency

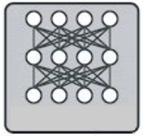
better sample efficiency

- model-based
- model-free
    - off-policy

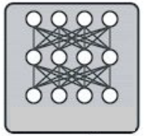
      can learns from data generated by a different policy
    - on-policy

      must learn from data generated by the current policy

off-policy RL



on-policy RL



Context

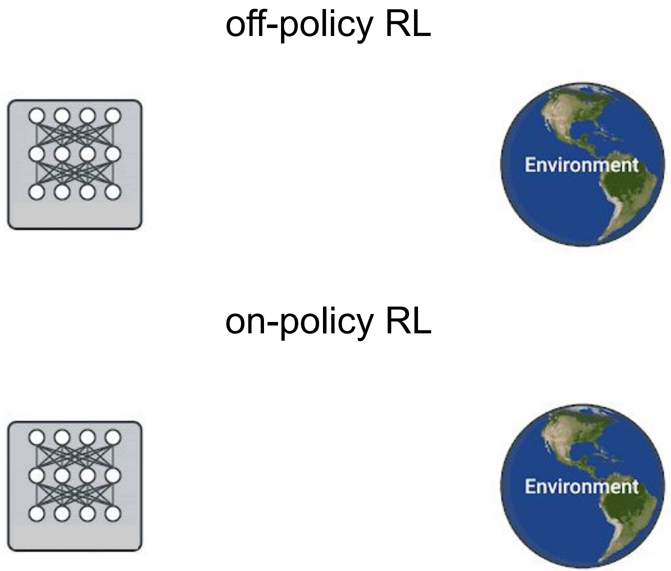
RL Ranking on Sample Efficiency

better sample efficiency

- model-based
- model-free
  - **off-policy**

    can learns from data generated by a different policy
  - on-policy

    must learn from data generated by the current policy



Context

Actor-Critic

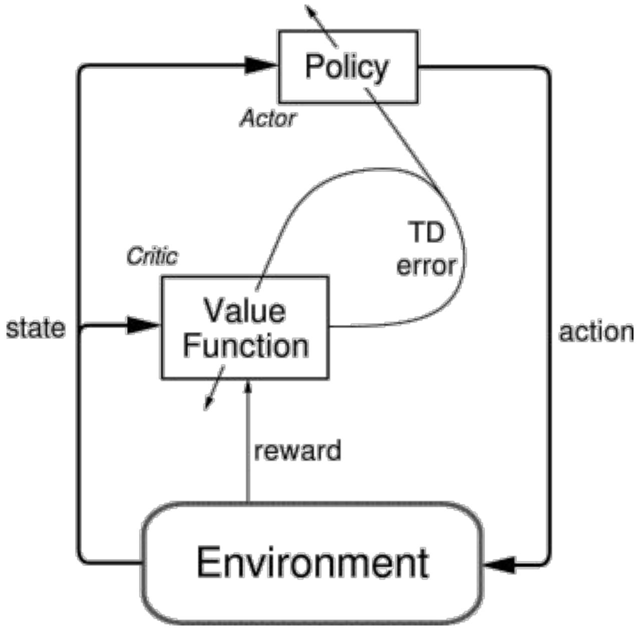
- Critic: estimates future return of state-action pair

$$Q^\pi(s,a) = \mathop{\mathbb{E}}_{\tau\sim\pi}[\textstyle\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t)|s=s_0,a=a_0]$$

$$Q^\pi(s,a) = \mathop{\mathbb{E}}_{s'\sim p, a'\sim\pi(\cdot|s')}[r(s,a) + \gamma Q^\pi(s',a')]$$

- Actor: adjusts policy to maximize critic's estimated future return

$$\pi(s) = \arg\max_a Q^\pi(s,a)$$



Context

Actor-Critic

- Critic: $Q^\pi(s,a) = \mathop{\mathbb{E}}_{s' \sim p, a' \sim \pi(\cdot|s')}[r(s,a) + \gamma Q^\pi(s',a')]$
- Actor: $\pi(s) = \arg\max_a Q^\pi(s,a)$

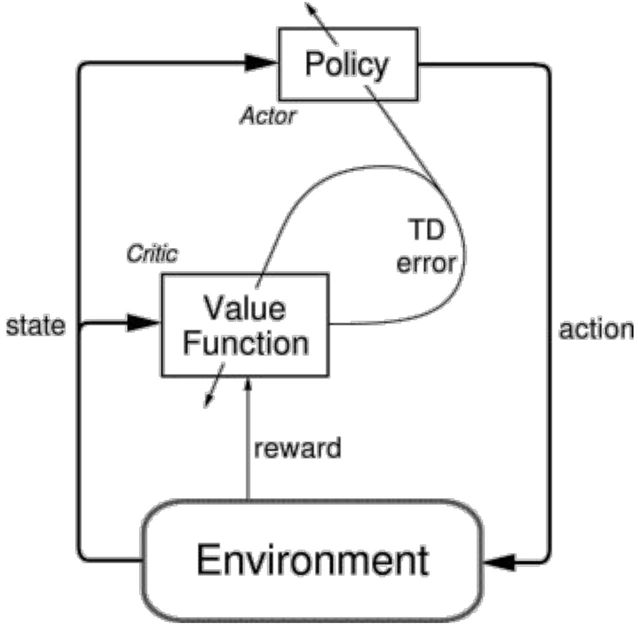
Deep Deterministic Policy Gradient (DDPG)

- Critic

$$L_Q = \mathop{\mathbb{E}}_{(s,a,r,s') \sim D}\left[\left(Q(s,a) - (r + \gamma Q(s', \pi(s'))\right)^2\right]$$

- Actor

$$L_\pi = -\mathop{\mathbb{E}}_{s \sim D}[Q(s, \pi(s))]$$



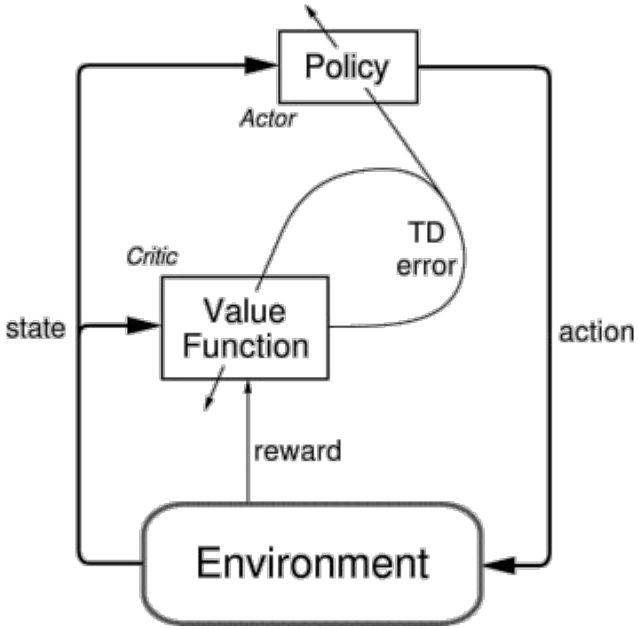
Context

Actor-Critic

- Critic: $Q^\pi(s,a) = \underset{s' \sim p, a' \sim \pi(\cdot|s')}{\mathbb{E}}[r(s,a) + \gamma Q^\pi(s',a')]$
- Actor: $\pi(s) = \arg\max_a Q^\pi(s,a)$

Deep Deterministic Policy Gradient (DDPG)

- Critic
$$L_Q = \underset{(s,a,r,s') \sim D}{\mathbb{E}}\left[\left(Q(s,a) - (r + \gamma Q(s', \pi(s')))\right)^2\right]$$
- Actor
$$L_\pi = -\underset{s \sim D}{\mathbb{E}}[Q(s, \pi(s))]$$
- Deterministic policy makes training unstable and brittle to hyperparameters.



Context

Standard RL

Reward: $r(s, a)$

Maximum Entropy RL

reward: $r(s, a) + \alpha \mathcal{H}(\pi(\cdot|s))$

Context

Standard RL

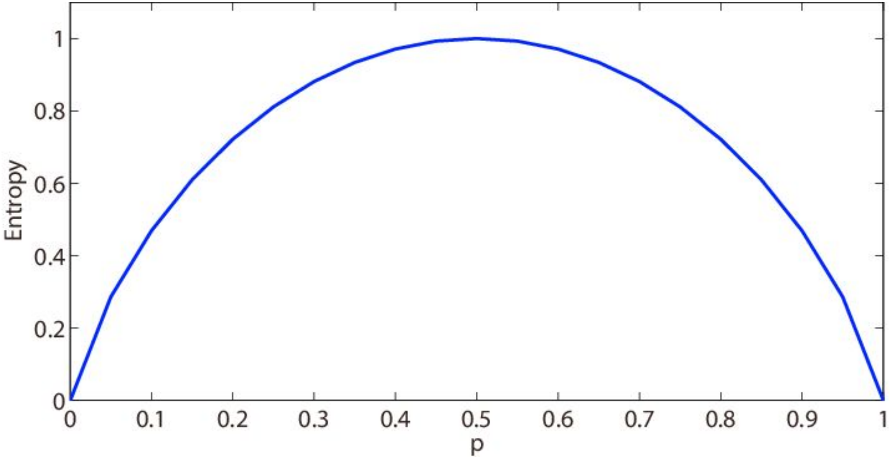
Reward: $r(s, a)$

Maximum Entropy RL

reward: $r(s, a) + \alpha \mathcal{H}(\pi(\cdot|s))$

Entropy $\mathcal{H}$: measure of uncertainty

$$\mathcal{H}(p) = -\int p(x) \log p(x) dx = -\mathbb{E}_{x \sim p}[\log p(x)]$$



Context

Standard RL

reward: $r(s, a)$

optimal policy:

$$\pi^* = \arg\max \sum_t \mathbb{E}[\gamma^t r(s_t, a_t)]$$

Maximum Entropy RL

reward: $r(s, a) + \alpha \mathcal{H}(\pi(\cdot|s))$

optimal policy:

$$\pi^* = \arg\max \sum_t \mathbb{E}[\gamma^t (r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot|s_t)))]$$

Context

Standard RL

reward: $r(s, a)$

optimal policy:

$$\pi^* = \arg\max \sum_t \mathbb{E}[\gamma^t r(s_t, a_t)]$$

Maximum Entropy RL

reward: $r(s, a) + \alpha\mathcal{H}(\pi(\cdot|s))$

optimal policy:

$$\pi^* = \arg\max \sum_t \mathbb{E}[\gamma^t (r(s_t, a_t) + \alpha\mathcal{H}(\pi(\cdot|s_t)))]$$

Why is it helpful?

- encourages exploration
- enables multi-modal action selection

Context

Standard RL

reward: $r(s, a)$

optimal policy:

$$\pi^* = \arg\max \sum_t \mathbb{E}[\gamma^t r(s_t, a_t)]$$

Maximum Entropy RL

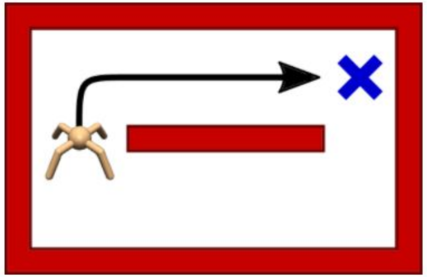
reward: $r(s, a) + \alpha \mathcal{H}(\pi(\cdot|s))$

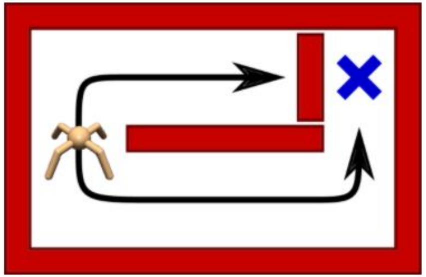
optimal policy:

$$\pi^* = \arg\max \sum_t \mathbb{E}[\gamma^t (r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot|s_t)))]$$

Why is it helpful?

- encourages exploration
- enables multi-modal action selection





Method

Actor-Critic (DDPG)

reward: $r(s, a)$

Q-value:

$$Q^\pi(s_0, a_0) = \mathop{\mathbb{E}}_{\tau \sim \pi}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$$

Soft Actor-Critic

reward: $r(s, a) + \alpha \mathcal{H}(\pi(\cdot|s))$

Q-value:

$$Q^\pi(s_0, a_0) = \mathop{\mathbb{E}}_{\tau \sim \pi}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) + \alpha \sum_{t=1}^{\infty} \gamma^t \mathcal{H}(\pi(\cdot|s_t))]$$

Method

Actor-Critic (DDPG)

reward: $r(s, a)$

Q-value:

$$Q^{\pi}(s_0, a_0) = \mathbb{E}_{\tau \sim \pi}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$$

Q-value updates:

$$Q^{\pi}(s, a) = \mathbb{E}_{\substack{s' \sim p \\ a' \sim \pi}} [r(s, a) + \gamma Q^{\pi}(s', a')]$$

Soft Actor-Critic

reward: $r(s, a) + \alpha \mathcal{H}(\pi(\cdot|s))$

Q-value:

$$Q^{\pi}(s_0, a_0) = \mathbb{E}_{\tau \sim \pi}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) + \alpha \sum_{t=1}^{\infty} \gamma^t \mathcal{H}(\pi(\cdot|s_t))]$$

Q-value updates:

$$Q^{\pi}(s, a) = \mathbb{E}_{\substack{s' \sim p \\ a' \sim \pi}} [r(s, a) + \gamma (Q^{\pi}(s', a') + \alpha \mathcal{H}(\pi(\cdot|s')))]$$

Method

Actor-Critic (DDPG)

reward: $r(s, a)$

Q-value:

$$Q^\pi(s_0, a_0) = \mathop{\mathbb{E}}_{\tau\sim\pi}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$$

Q-value updates:

$$Q^\pi(s, a) = \mathop{\mathbb{E}}_{\substack{s'\sim p \\ a'\sim\pi}} [r(s, a) + \gamma Q^\pi(s', a')]$$

policy improvement:

$$\pi_{\text{new}} = \arg\max_\pi Q^{\pi_{\text{old}}}(s_t, \pi(\cdot|s_t))$$

Soft Actor-Critic

reward: $r(s, a) + \alpha\mathcal{H}(\pi(\cdot|s))$

Q-value:

$$Q^\pi(s_0, a_0) = \mathop{\mathbb{E}}_{\tau\sim\pi}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) + \alpha \sum_{t=1}^{\infty} \gamma^t \mathcal{H}(\pi(\cdot|s_t))]$$

Q-value updates:

$$Q^\pi(s, a) = \mathop{\mathbb{E}}_{\substack{s'\sim p \\ a'\sim\pi}} [r(s, a) + \gamma\,(Q^\pi(s', a') + \alpha\mathcal{H}(\pi(\cdot|s')))]$$

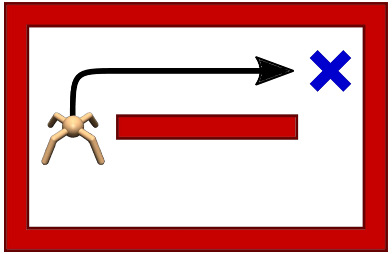
policy improvement:

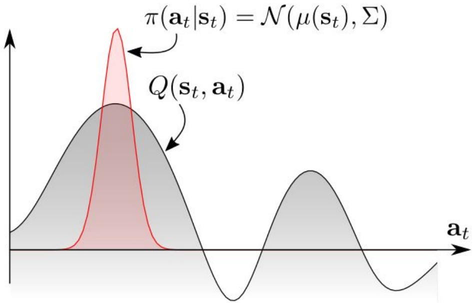
$$\pi_{\text{new}} = \arg\min_\pi D_{\text{KL}}\left(\pi(\cdot|s_t) \middle|\middle| \frac{\exp(\frac{1}{\alpha} Q^{\pi_{\text{old}}}(s_t, \cdot))}{Z^{\pi_{\text{old}}}(s_t)}\right)$$

Method

Actor-Critic (DDPG)

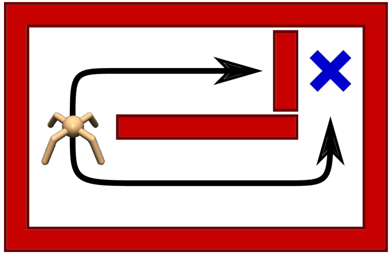
$$\pi_{\text{new}} = \arg\max_{\pi} Q^{\pi_{\text{old}}}(s_t, \pi(\cdot|s_t))$$

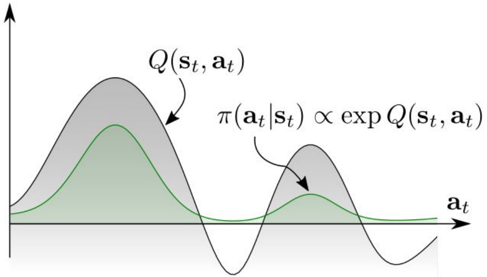




Soft Actor-Critic

$$\pi_{\text{new}} = \arg\min_{\pi} D_{\text{KL}}\left(\pi(\cdot|s_t) \middle\| \frac{\exp(\frac{1}{\alpha}Q^{\pi_{\text{old}}}(s_t, \cdot))}{Z^{\pi_{\text{old}}}(s_t)}\right)$$





Method

Proof of Convergence

Assumption: finite state and action space $|S| < \infty, |A| < \infty$

1. If we update Q-value as follows, it will converge to Q^π as $k \to \infty$.

$$Q^{k+1}(s,a) = \mathop{\mathbb{E}}_{\substack{s' \sim p \\ a' \sim \pi}} \left[r(s,a) + \gamma \left(Q^k(s',a') + \alpha \mathcal{H}(\pi(\cdot|s')) \right) \right]$$

Method

Proof of Convergence

Assumption: finite state and action space $|S| < \infty, |A| < \infty$

1. If we update Q-value as follows, it will converge to Q^π as $k \to \infty$.

$$Q^{k+1}(s,a) = \mathop{\mathbb{E}}_{\substack{s' \sim p \\ a' \sim \pi}} \left[r(s,a) + \gamma \left(Q^k(s', a') + \alpha \mathcal{H}(\pi(\cdot|s')) \right) \right]$$

2. If we update the policy as follows, then $Q^{\pi_{\text{new}}}(s_t, a_t) \geq Q^{\pi_{\text{old}}}(s_t, a_t), \forall (s_t, a_t) \in \mathcal{S} \times \mathcal{A}$.

$$\pi_{\text{new}} = \arg\min_{\pi} D_{\text{KL}} \left(\pi(\cdot|s_t) \middle|\middle| \frac{\exp(\frac{1}{\alpha} Q^{\pi_{\text{old}}}(s_t, \cdot))}{Z^{\pi_{\text{old}}}(s_t)} \right)$$

Method

Proof of Convergence

Assumption: finite state and action space $|S| < \infty, |A| < \infty$

1. If we update Q-value as follows, it will converge to Q^π as $k \to \infty$.

$$Q^{k+1}(s,a) = \mathbb{E}_{\substack{s' \sim p \\ a' \sim \pi}} \left[r(s,a) + \gamma \left(Q^k(s',a') + \alpha \mathcal{H}(\pi(\cdot|s')) \right) \right]$$

2. If we update the policy as follows, then $Q^{\pi_{\text{new}}}(s_t,a_t) \geq Q^{\pi_{\text{old}}}(s_t,a_t), \forall (s_t,a_t) \in \mathcal{S} \times \mathcal{A}$.

$$\pi_{\text{new}} = \arg\min_{\pi} D_{\text{KL}} \left(\pi(\cdot|s_t) \middle|\middle| \frac{\exp(\frac{1}{\alpha} Q^{\pi_{\text{old}}}(s_t,\cdot))}{Z^{\pi_{\text{old}}}(s_t)} \right)$$

3. If we repeat step 1 and 2, we will find the optimal policy π^* such that $Q^{\pi^*}(s_t,a_t) \geq Q^{\pi}(s_t,a_t)$ for any policy π and $\forall (s_t,a_t) \in \mathcal{S} \times \mathcal{A}$.

Method

Proof of Convergence: Are they realistic?

Assumption: finite state and action space $|S| < \infty, |A| < \infty$

1. If we update Q-value as follows, it will converge to Q^π as $k \to \infty$.

$$Q^{k+1}(s,a) = \mathop{\mathbb{E}}_{\substack{s' \sim p \\ a' \sim \pi}} \left[r(s,a) + \gamma \left(Q^k(s', a') + \alpha \mathcal{H}(\pi(\cdot|s')) \right) \right]$$

2. If we update the policy as follows, then $Q^{\pi_{\text{new}}}(s_t, a_t) \geq Q^{\pi_{\text{old}}}(s_t, a_t), \forall (s_t, a_t) \in \mathcal{S} \times \mathcal{A}$.

$$\pi_{\text{new}} = \arg\min_{\pi} D_{\text{KL}} \left(\pi(\cdot|s_t) \middle|\middle| \frac{\exp(\frac{1}{\alpha} Q^{\pi_{\text{old}}}(s_t, \cdot))}{Z^{\pi_{\text{old}}}(s_t)} \right)$$

3. If we repeat step 1 and 2, we will find the optimal policy π^* such that $Q^{\pi^*}(s_t, a_t) \geq Q^\pi(s_t, a_t)$ for any policy π and $\forall (s_t, a_t) \in \mathcal{S} \times \mathcal{A}$.

Method

Proof of Convergence: Are they realistic?

Assumption: finite state and action space $|S| < \infty, |A| < \infty$

1. If we update Q-value as follows, it will converge to Q^π as $k \to \infty$.

$$Q^{k+1}(s,a) = \mathop{\mathbb{E}}_{\substack{s' \sim p \\ a' \sim \pi}} \left[r(s,a) + \gamma \left(Q^k(s',a') + \alpha \mathcal{H}(\pi(\cdot|s')) \right) \right]$$

2. If we update the policy as follows, then $Q^{\pi_{\text{new}}}(s_t, a_t) \geq Q^{\pi_{\text{old}}}(s_t, a_t), \forall (s_t, a_t) \in \mathcal{S} \times \mathcal{A}$.

$$\pi_{\text{new}} = \arg\min_{\pi} D_{\text{KL}} \left(\pi(\cdot|s_t) \middle\| \frac{\exp(\frac{1}{\alpha} Q^{\pi_{\text{old}}}(s_t, \cdot))}{Z^{\pi_{\text{old}}}(s_t)} \right)$$

3. If we repeat step 1 and 2, we will find the optimal policy π^* such that $Q^{\pi^*}(s_t, a_t) \geq Q^\pi(s_t, a_t)$ for any policy π and $\forall (s_t, a_t) \in \mathcal{S} \times \mathcal{A}$. For how many step? Possibly a lot.

Method

Proof of Convergence: Are they realistic?

Assumption: finite state and action space $|S| < \infty, |A| < \infty$

1. If we update Q-value as follows, it will converge to Q^π as $k \to \infty$.

$$Q^{k+1}(s,a) = \mathop{\mathbb{E}}_{\substack{s' \sim p \\ a' \sim \pi}} \left[r(s,a) + \gamma \left(Q^k(s',a') + \alpha \mathcal{H}(\pi(\cdot|s')) \right) \right]$$

2. If we update the policy as follows, then $Q^{\pi_\text{new}}(s_t,a_t) \geq Q^{\pi_\text{old}}(s_t,a_t), \forall (s_t,a_t) \in \mathcal{S} \times \mathcal{A}$.

$$\pi_\text{new} = \arg\min_{\pi} D_\text{KL}\left(\pi(\cdot|s_t) \middle\| \frac{\exp(\frac{1}{\alpha} Q^{\pi_\text{old}}(s_t,\cdot))}{Z^{\pi_\text{old}}(s_t)} \right)$$

3. If we repeat step 1 and 2, we will find the optimal policy π^* such that $Q^{\pi^*}(s_t,a_t) \geq Q^\pi(s_t,a_t)$ for any policy π and $\forall (s_t,a_t) \in \mathcal{S} \times \mathcal{A}$. For how many step? Possibly a lot.

Not applicable to many robotics problems (continuous state/action) and not computationally tractable.

Method

Implementation (Practical Approximation)

1. critic training

- convergence → gradient descent

$$Q^{k+1}(s,a) = \mathop{\mathbb{E}}_{\substack{s' \sim p \\ a' \sim \pi}} \left[r(s,a) + \gamma \left(Q^k(s',a') + \alpha \mathcal{H}(\pi(\cdot|s')) \right) \right], k \to \infty$$

$$L_Q = \mathbb{E}_{(s,a,r)\sim D} \left[\left(Q(s,a) - \left(r + \gamma \mathop{\mathbb{E}}_{s' \sim p} \left[\mathop{\mathbb{E}}_{a' \sim \pi(\cdot|s')} [Q(s',a')] + \alpha \mathcal{H}(\pi(\cdot|s')) \right] \right) \right)^2 \right]$$

Method

Implementation (Practical Approximation)

1. critic training
   - convergence → gradient descent
   - remove expectation
     - next action

$$Q^{k+1}(s,a) = \mathop{\mathbb{E}}_{\substack{s' \sim p \\ a' \sim \pi}} \left[r(s,a) + \gamma \left(Q^k(s',a') + \alpha \mathcal{H}(\pi(\cdot|s')) \right) \right], k \to \infty$$

$$\begin{aligned} L_Q =& \mathbb{E}_{(s,a,r)\sim D} \left[\left(Q(s,a) - \left(r + \gamma \mathop{\mathbb{E}}_{s' \sim p} \left[\mathop{\mathbb{E}}_{a' \sim \pi(\cdot|s')} [Q(s',a')] + \alpha \mathcal{H}(\pi(\cdot|s')) \right] \right) \right)^2 \right] \\ =& \mathbb{E}_{(s,a,r)\sim D} \left[\left(Q(s,a) - \left(r + \gamma \mathop{\mathbb{E}}_{s' \sim p} [Q(s', {\color{red}a'}) + \alpha \mathcal{H}(\pi(\cdot|s'))] \right) \right)^2 \right], {\color{red}a' \sim \pi(\cdot|s')} \end{aligned}$$

Method

Implementation (Practical Approximation)

1. critic training
   - convergence → gradient descent
   - remove expectation
     - next action
     - entropy

       $\mathcal{H}(p) = -\mathbb{E}_{x \sim p}[\log p(x)]$

$$Q^{k+1}(s,a) = \mathop{\mathbb{E}}_{\substack{s' \sim p \\ a' \sim \pi}} \left[r(s,a) + \gamma \left(Q^k(s',a') + \alpha \mathcal{H}(\pi(\cdot|s')) \right) \right], k \to \infty$$

$$\begin{aligned} L_Q =& \mathbb{E}_{(s,a,r)\sim D} \left[\left(Q(s,a) - \left(r + \gamma \mathop{\mathbb{E}}_{s' \sim p} \left[\mathop{\mathbb{E}}_{a' \sim \pi(\cdot|s')} [Q(s',a')] + \alpha \mathcal{H}(\pi(\cdot|s')) \right] \right) \right)^2 \right] \\ =& \mathbb{E}_{(s,a,r)\sim D} \left[\left(Q(s,a) - \left(r + \gamma \mathop{\mathbb{E}}_{s' \sim p} [Q(s',a') + \alpha \mathcal{H}(\pi(\cdot|s'))] \right) \right)^2 \right], a' \sim \pi(\cdot|s') \\ =& \mathbb{E}_{(s,a,r)\sim D} \left[\left(Q(s,a) - \left(r + \gamma \mathop{\mathbb{E}}_{s' \sim p} [Q(s',a') - \alpha \log \pi(a'|s')] \right) \right)^2 \right] \end{aligned}$$

Method

Implementation (Practical Approximation)

1. critic training
    - convergence → gradient descent
    - remove expectation
        - next action
        - entropy

          $\mathcal{H}(p) = -\mathbb{E}_{x\sim p}[\log p(x)]$
        - next state

$$Q^{k+1}(s,a) = \underset{\substack{s'\sim p\\ a'\sim\pi}}{\mathbb{E}}\left[r(s,a) + \gamma\left(Q^k(s',a') + \alpha\mathcal{H}(\pi(\cdot|s'))\right)\right], k \to \infty$$

$$\begin{aligned}
L_Q &= \mathbb{E}_{(s,a,r)\sim D}\left[\left(Q(s,a) - \left(r + \gamma \underset{s'\sim p}{\mathbb{E}}\left[\underset{a'\sim\pi(\cdot|s')}{\mathbb{E}}[Q(s',a')] + \alpha\mathcal{H}(\pi(\cdot|s'))\right]\right)\right)^2\right] \\
&= \mathbb{E}_{(s,a,r)\sim D}\left[\left(Q(s,a) - \left(r + \gamma \underset{s'\sim p}{\mathbb{E}}[Q(s',a') + \alpha\mathcal{H}(\pi(\cdot|s'))]\right)\right)^2\right], a' \sim \pi(\cdot|s') \\
&= \mathbb{E}_{(s,a,r)\sim D}\left[\left(Q(s,a) - \left(r + \gamma \underset{s'\sim p}{\mathbb{E}}[Q(s',a') - \alpha\log\pi(a'|s')]\right)\right)^2\right] \\
&= \mathbb{E}_{(s,a,r,s')\sim D}\left[(Q(s,a) - (r + \gamma Q(s',a') - \alpha\log\pi(a'|s')))^2\right]
\end{aligned}$$

Method

Implementation (Practical Approximation)

2. actor

training

- convergence → gradient descent
- rewrite KL divergence

$$D_{\mathrm{KL}}(p||q) = \mathop{\mathbb{E}}_{x\sim p(\cdot)}\left[\log \frac{p(x)}{q(x)}\right]$$

$$\pi_{\text{new}} = \arg\min_{\pi} D_{\mathrm{KL}}\left(\pi(\cdot|s_t)\Big|\Big|\frac{\exp(\frac{1}{\alpha}Q^{\pi_{\text{old}}}(s_t,\cdot))}{Z^{\pi_{\text{old}}}(s_t)}\right)$$

$$L_\pi = \mathop{\mathbb{E}}_{s\sim D}\left[\mathop{\mathbb{E}}_{a'\sim\pi(\cdot|s')}\left[\log \frac{\pi(a'|s')}{\exp(\frac{1}{\alpha}Q(s',a'))/Z^\pi}\right]\right]$$

Method

Implementation (Practical Approximation)

2. actor

training

- convergence → gradient descent
- rewrite KL divergence

$$\pi_{\text{new}} = \arg\min_{\pi} D_{\text{KL}} \left(\pi(\cdot|s_t) \middle|\middle| \frac{\exp(\frac{1}{\alpha} Q^{\pi_{\text{old}}}(s_t, \cdot))}{Z^{\pi_{\text{old}}}(s_t)} \right)$$

$$\begin{aligned} L_\pi &= \mathop{\mathbb{E}}_{s\sim D}\left[\mathop{\mathbb{E}}_{a'\sim\pi(\cdot|s')}\left[\log \frac{\pi(a'|s')}{\exp(\frac{1}{\alpha}Q(s',a'))/Z^\pi} \right]\right] \\ &= \mathop{\mathbb{E}}_{s\sim D}\left[\mathop{\mathbb{E}}_{a'\sim\pi(\cdot|s')}\left[\log \pi(a'|s') - \frac{1}{\alpha}Q(s',a')) + \log Z^\pi \right]\right] \end{aligned}$$

Method

Implementation (Practical Approximation)

2. actor

training

- convergence → gradient descent
- rewrite KL divergence
- scale loss by α and omit constant

$$\pi_{\text{new}} = \arg\min_{\pi} D_{\text{KL}} \left(\pi(\cdot|s_t) \middle|\middle| \frac{\exp(\frac{1}{\alpha} Q^{\pi_{\text{old}}}(s_t, \cdot))}{Z^{\pi_{\text{old}}}(s_t)} \right)$$

$$\begin{aligned} L_\pi &= \mathop{\mathbb{E}}_{s \sim D} \left[\mathop{\mathbb{E}}_{a' \sim \pi(\cdot|s')} \left[\log \frac{\pi(a'|s')}{\exp(\frac{1}{\alpha} Q(s', a'))/Z^\pi} \right] \right] \\ &= \mathop{\mathbb{E}}_{s \sim D} \left[\mathop{\mathbb{E}}_{a' \sim \pi(\cdot|s')} \left[\log \pi(a'|s') - \frac{1}{\alpha} Q(s', a')) + \log Z^\pi \right] \right] \\ &= \mathop{\mathbb{E}}_{s \sim D} \left[\mathop{\mathbb{E}}_{a' \sim \pi(\cdot|s')} [\alpha \log \pi(a'|s') - Q(s', a'))] \right] \end{aligned}$$

Method

Implementation (Practical Approximation)

2. actor

   training

   - convergence → gradient descent
   - rewrite KL divergence
   - scale loss by α and omit constant
   - remove expectation over action

$$\pi_{\text{new}} = \arg\min_{\pi} D_{\text{KL}} \left(\pi(\cdot|s_t) \middle\| \frac{\exp(\frac{1}{\alpha} Q^{\pi_{\text{old}}}(s_t,\cdot))}{Z^{\pi_{\text{old}}}(s_t)} \right)$$

$$\begin{aligned} L_\pi &= \mathop{\mathbb{E}}_{s\sim D}\left[\mathop{\mathbb{E}}_{a'\sim\pi(\cdot|s')}\left[\log \frac{\pi(a'|s')}{\exp(\frac{1}{\alpha}Q(s',a'))/Z^\pi}\right]\right] \\ &= \mathop{\mathbb{E}}_{s\sim D}\left[\mathop{\mathbb{E}}_{a'\sim\pi(\cdot|s')}\left[\log \pi(a'|s') - \frac{1}{\alpha}Q(s',a')) + \log Z^\pi\right]\right] \\ &= \mathop{\mathbb{E}}_{s\sim D}\left[\mathop{\mathbb{E}}_{a'\sim\pi(\cdot|s')}[\alpha \log \pi(a'|s') - Q(s',a'))]\right] \\ &= \mathop{\mathbb{E}}_{s\sim D}[\alpha \log \pi(a'|s') - Q(s',a'))], a' \sim \pi(\cdot|s') \end{aligned}$$

Method

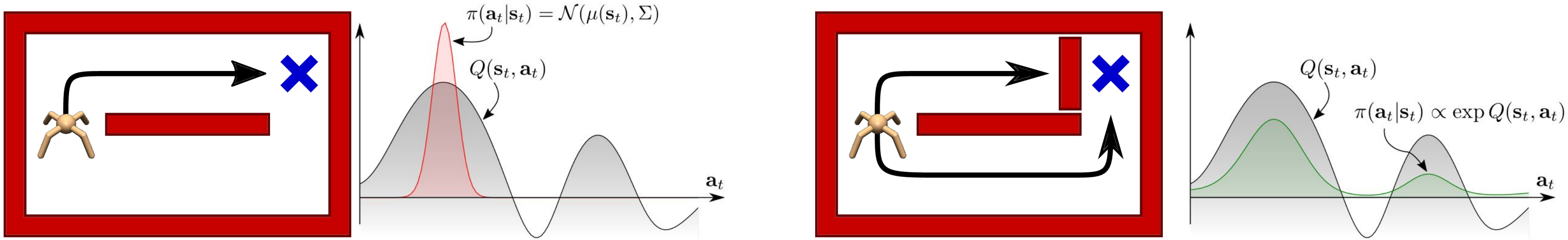
Implementation (Practical Approximation)

2. actor

training: gradient descent with $L_\pi = \underset{s\sim D}{\mathbb{E}}[\alpha \log \pi(a'|s') - Q(s', a'))], a' \sim \pi(\cdot|s')$

design choice: policy as normal distribution

- but normal distribution is unimodal, it loses the declared multi-modal advantages.



Method

Automatic Entropy Adjustment

Choosing the optimal α is not trivial

- Recall for maximum entropy RL, the reward is $r(s, a) + \alpha\mathcal{H}(\pi(\cdot|s))$.

Method

Automatic Entropy Adjustment

Choosing the optimal α is not trivial

- Recall for maximum entropy RL, the reward is $r(s, a) + \alpha\mathcal{H}(\pi(\cdot|s))$.
- So α depends on the magnitude of r, but r can vary a lot across
  - different tasks
  - different policies as the policy gets improved

Method

Automatic Entropy Adjustment

Choosing the optimal α is not trivial

- Recall for maximum entropy RL, the reward is $r(s, a) + \alpha\mathcal{H}(\pi(\cdot|s))$.
- So α depends on the magnitude of r, but r can vary a lot across
    - different tasks
    - different policies as the policy gets improved
- Ideal policy entropy should be
    - stochastic enough for exploration in uncertain regions
    - deterministic enough for performance in learned regions

Method

Automatic Entropy Adjustment

Choosing the optimal α is not trivial

- Recall for maximum entropy RL, the reward is $r(s,a) + \alpha\mathcal{H}(\pi(\cdot|s))$.
- So α depends on the magnitude of r, but r can vary a lot across
    - different tasks
    - different policies as the policy gets improved
- Ideal policy entropy should be
    - stochastic enough for exploration in uncertain regions
    - deterministic enough for performance in learned regions
- Only constrain the **average** entropy across states

Method

Automatic Entropy Adjustment

Only constrain the **average** entropy across states

$$\begin{aligned} L_\alpha &= \mathop{\mathbb{E}}_{s\sim D}\left[\alpha(\mathcal{H}(\pi(\cdot|s)) - \bar{\mathcal{H}})\right] \\ &= \mathop{\mathbb{E}}_{s\sim D}\left[\alpha(-\log \pi(a|s) - \bar{\mathcal{H}})\right], a \sim \pi(\cdot|s) \end{aligned}$$

- $\bar{\mathcal{H}}$: target entropy, α always > 0
- increase α if $\mathcal{H}(\pi(\cdot|s')) < \bar{\mathcal{H}}$, decrease otherwise.

Method

Overall Algorithm

Input: θ_1, θ_2, ϕ ▷ Initial parameters
$\bar{\theta}_1 \leftarrow \theta_1, \bar{\theta}_2 \leftarrow \theta_2$ ▷ Initialize target network weights
$\mathcal{D} \leftarrow \emptyset$ ▷ Initialize an empty replay pool
for each iteration **do**
  for each environment step **do**
    $\mathbf{a}_t \sim \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)$ ▷ Sample action from the policy
    $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ ▷ Sample transition from the environment
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1})\}$ ▷ Store the transition in the replay pool
  end for
  for each gradient step **do**
    $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$ ▷ Update the Q-function parameters
    $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$ ▷ Update policy weights
    $\alpha \leftarrow \alpha - \lambda \hat{\nabla}_\alpha J(\alpha)$ ▷ Adjust temperature
    $\bar{\theta}_i \leftarrow \tau\theta_i + (1 - \tau)\bar{\theta}_i$ for $i \in \{1, 2\}$ ▷ Update target network weights
  end for
end for
Output: θ_1, θ_2, ϕ ▷ Optimized parameters

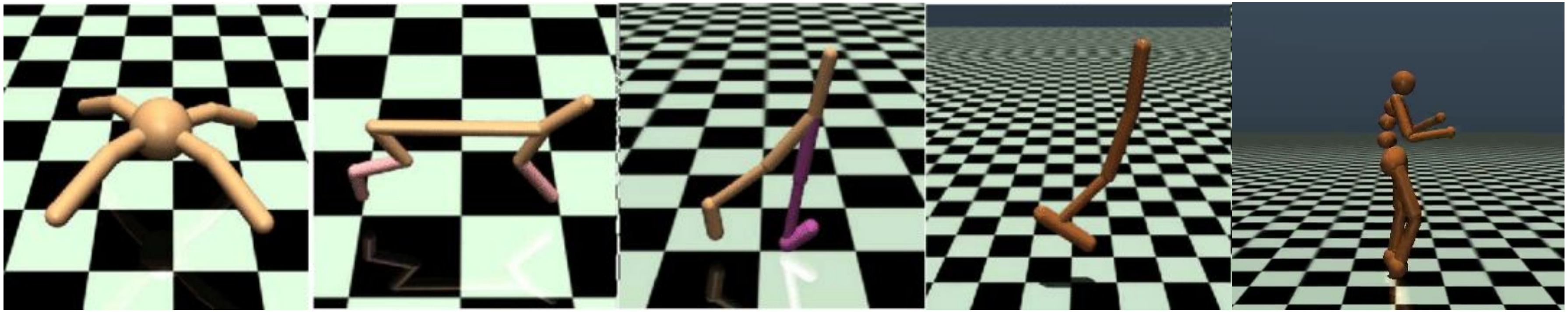
Experiment

Simulated Benchmarks

State: joint value

Action: joint torque

Metric: average return



Ant Half Cheetah Walker Hopper Humanoid

Experiment

Baselines

- SAC with learned α
- SAC with fixed α
- DDPG (off-policy, deterministic policy)
- TD3 (DDPG with engineering improvements)
- PPO (on-policy)

Experiment

Baselines

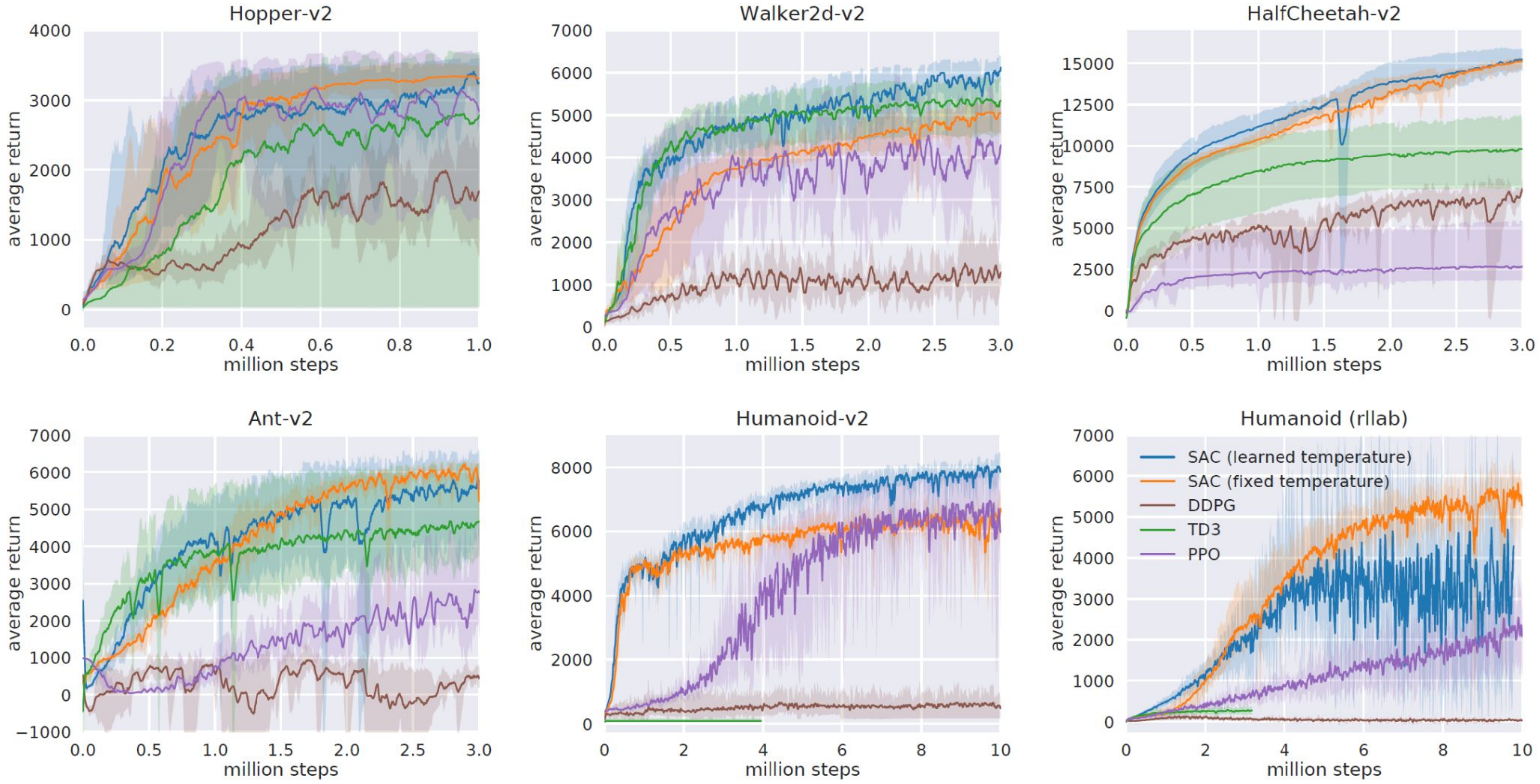
- SAC with learned α
- SAC with fixed α
- DDPG (off-policy, deterministic policy)
- TD3 (DDPG with engineering improvements)
- PPO (on-policy)

Hypotheses

Compared to baselines, if SAC has better

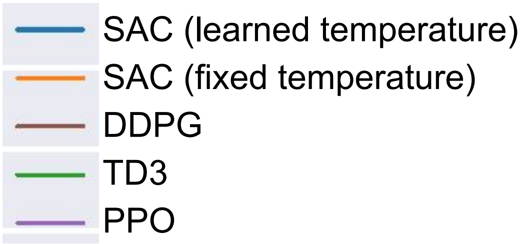
- sample-efficiency: learning speed + final performance
- stability: performance on hard tasks where hyperparameters tuning is challenging

Experiment



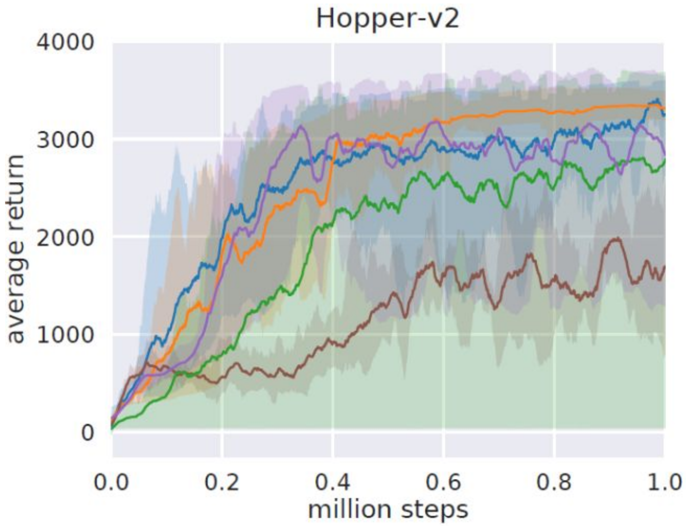
Hopper-v2
Walker2d-v2
HalfCheetah-v2
Ant-v2
Humanoid-v2
Humanoid (rllab)
average return
million steps
SAC (learned temperature)
SAC (fixed temperature)
DDPG
TD3
PPO

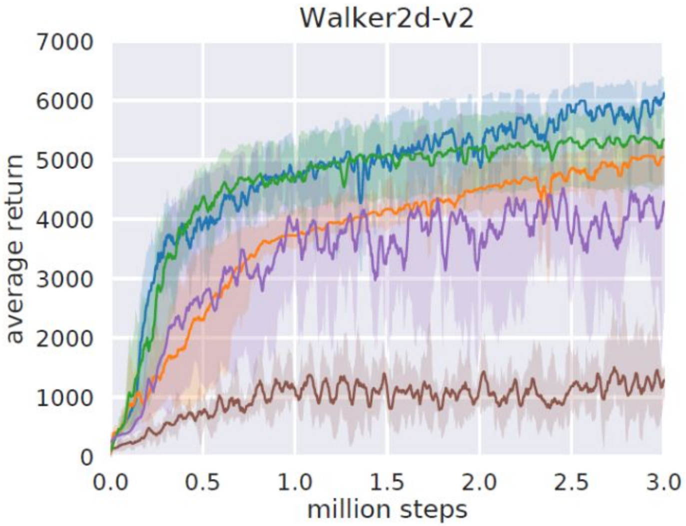
Experiment



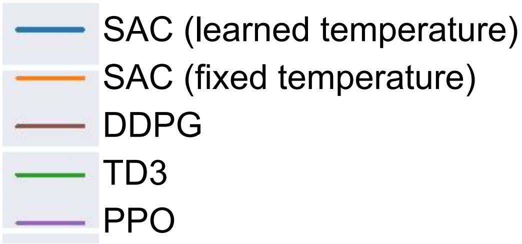
Simulated Benchmarks

- Easy tasks (hopper, walker)
  - all algorithm performs comparably except for DDPG



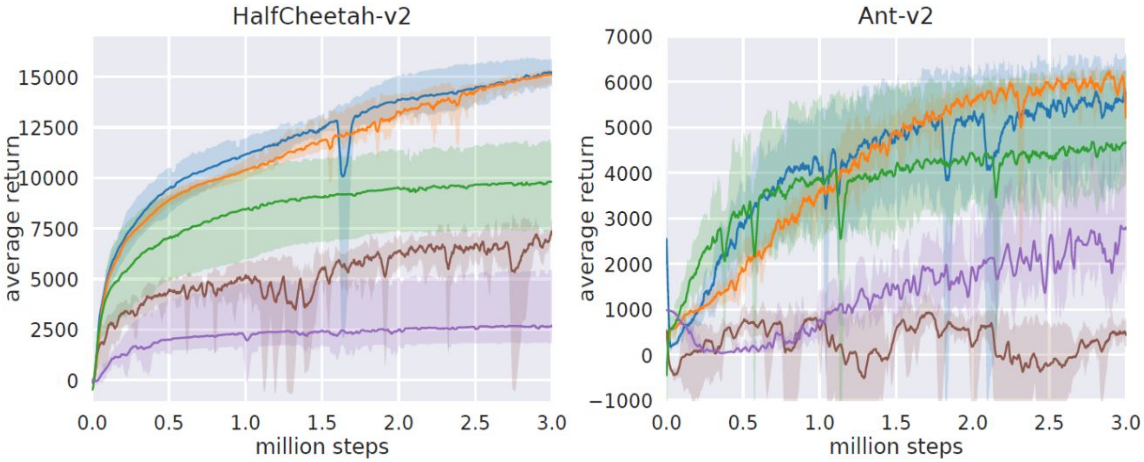


Experiment

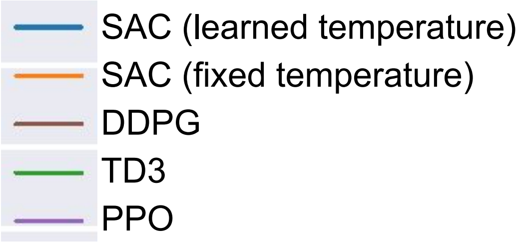


Simulated Benchmarks

- Normal tasks (half cheetah, ant)
  - SAC > TD3 > DDPG & PPO in both learning speed and final performance

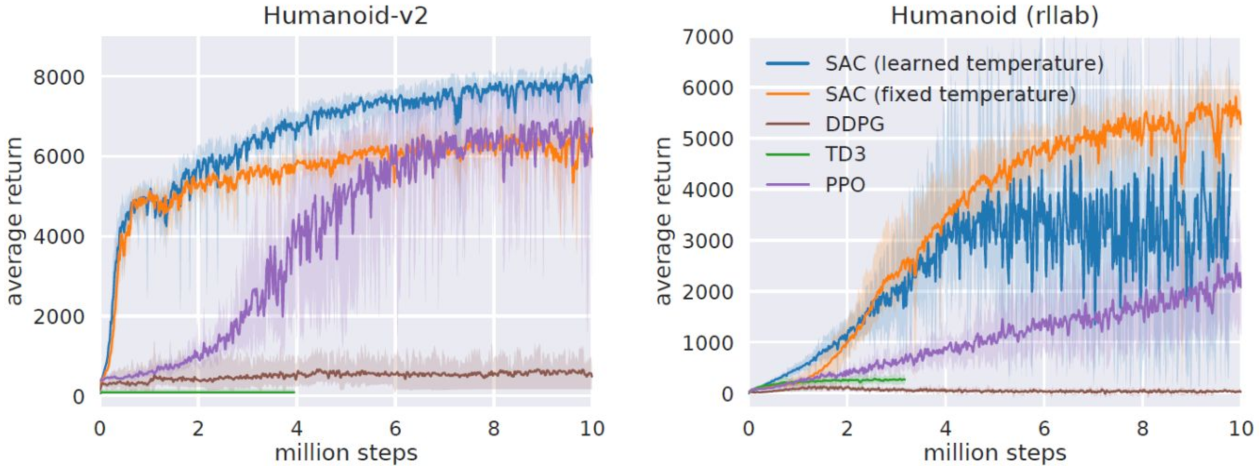


Experiment

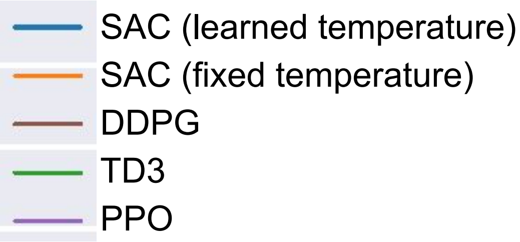


Simulated Benchmarks

- Hard tasks (humanoid)
  - DDPG & its variant TD3 fail to learn
  - SAC learns much faster than PPO

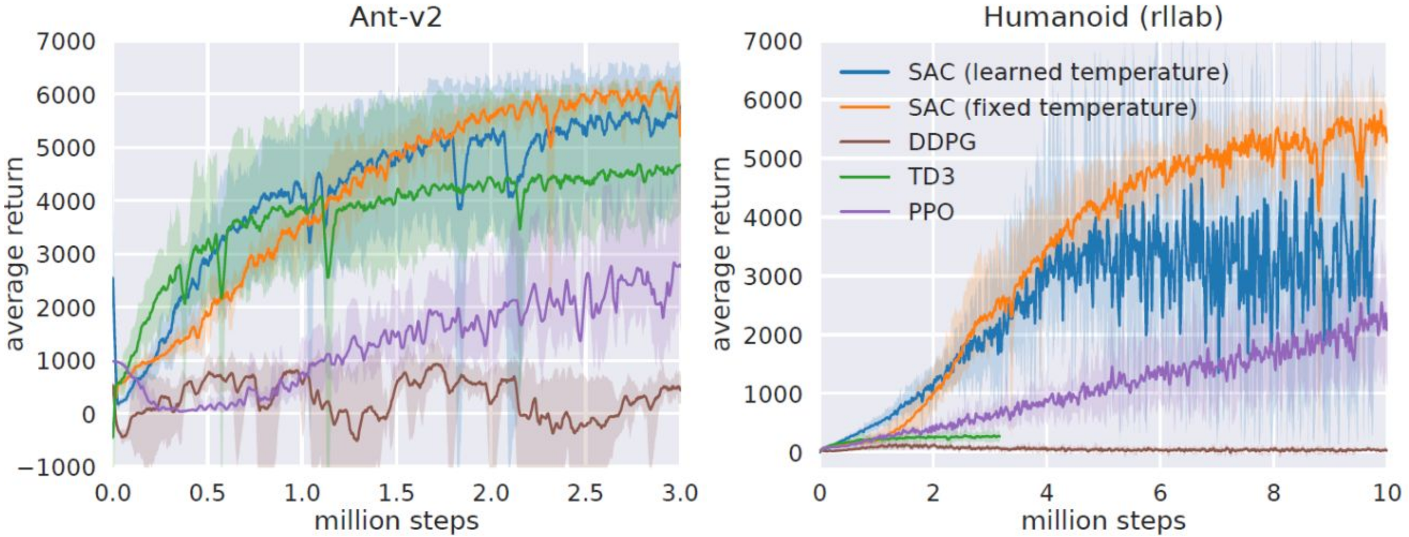


Experiment



Simulated Benchmarks

- Larger variance with the automatic temperature adjustment



Experiment

Real World Quadrupedal Locomotion

State: low-dimensional

Challenges: sample efficiency & generalization to unseen environments

Training: 160k steps (2 hours)

Experiment

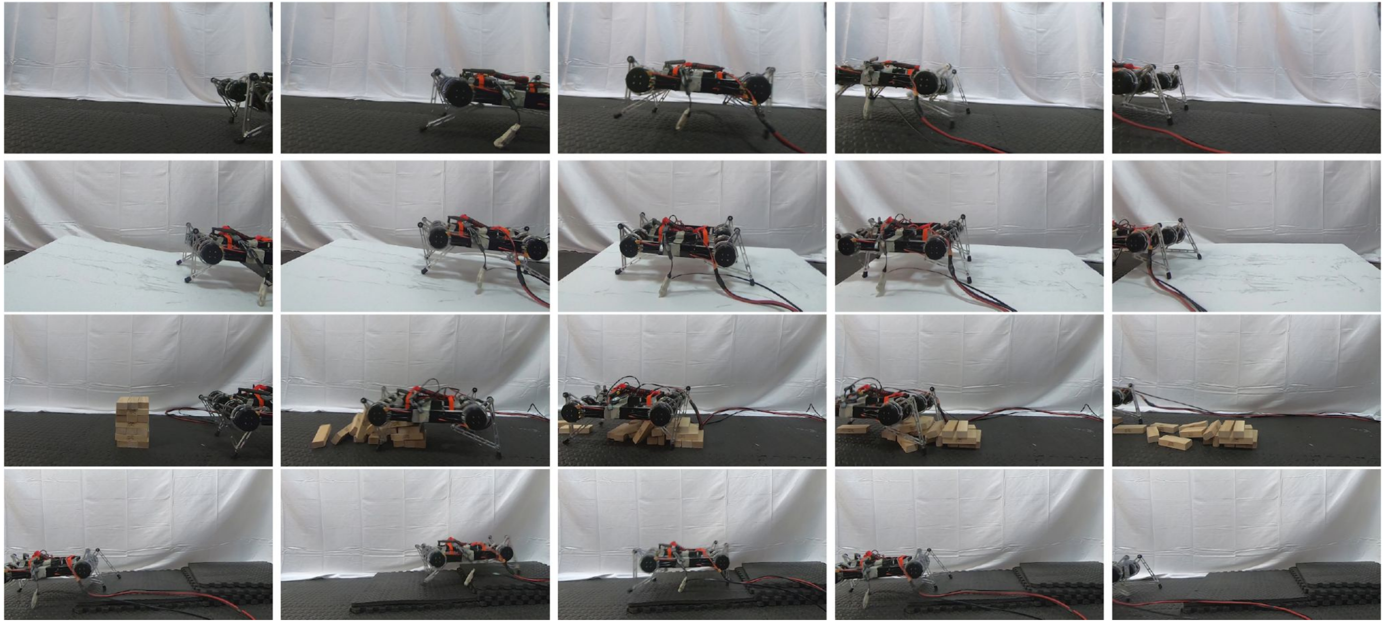
Real World Quadrupedal Locomotion

Train on Flat

Test on Slope

Test with Obstacles

Test with Stairs



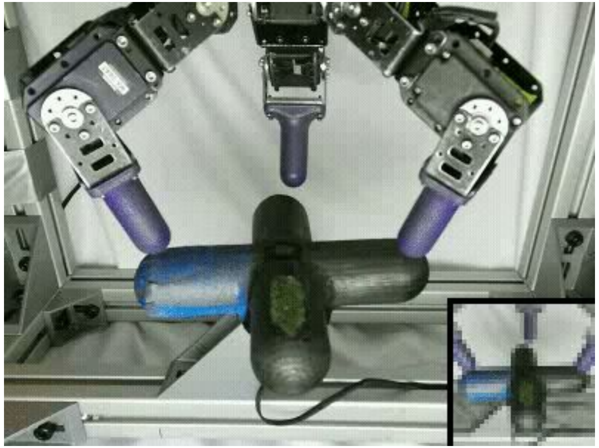
Experiment

Real World Manipulation

State: hand joint angles + image / ground truth valve position

Challenges: precepting the valve position from images

Comparison (using ground truth valve position): SAC (3 hrs) vs PPO (7.4 hrs)



Limitations

- Multi-modal policy
  - Though it is declared that maximum entropy RL can benefit from multi-modal policy, SAC chooses to use a unimodal policy (normal distribution).
  - All experiments don’t require multi-modal behaviors to finish.
- Hyperparameter tuning
  - Target entropy $\bar{\mathcal{H}}$ brings a new hyperparameter to tune.
  - Average entropy constraints do not provide the desired exploration + exploitation balance.
  - For manipulation tasks that require accurate control, even averaged entropy regularization still hurts the performance.

Future Work and Extended Readings

- Learn multi-modal policy rather than unimodal policy
    - Reinforcement Learning with Deep Energy-Based Policies
- Improve sample-efficiency with auxiliary tasks
    - Improving Sample Efficiency in Model-Free Reinforcement Learning from Images
    - CURL: Contrastive Unsupervised Representations for Reinforcement Learning
- Improve generalizability by learning tasks-relevant features
    - Learning Invariant Representations for Reinforcement Learning without Reconstruction
    - Learning Task Informed Abstractions

Summary

- Problem: sample-efficient RL with automatic hyperparameter tuning
- Limitations of prior work:
  - On-policy RL is sample-inefficient
  - DDPG that uses deterministic policy is brittle to hyperparameters
- Key insights of the proposed work
  - Maximal entropy RL encourages exploration and robust to environments & hyperparameters.
  - Use the average entropy across states as regularization.
- State-of-the-art sample efficiency and generalizabilities on simulated and real world tasks.