Soft Actor-Critic

Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor

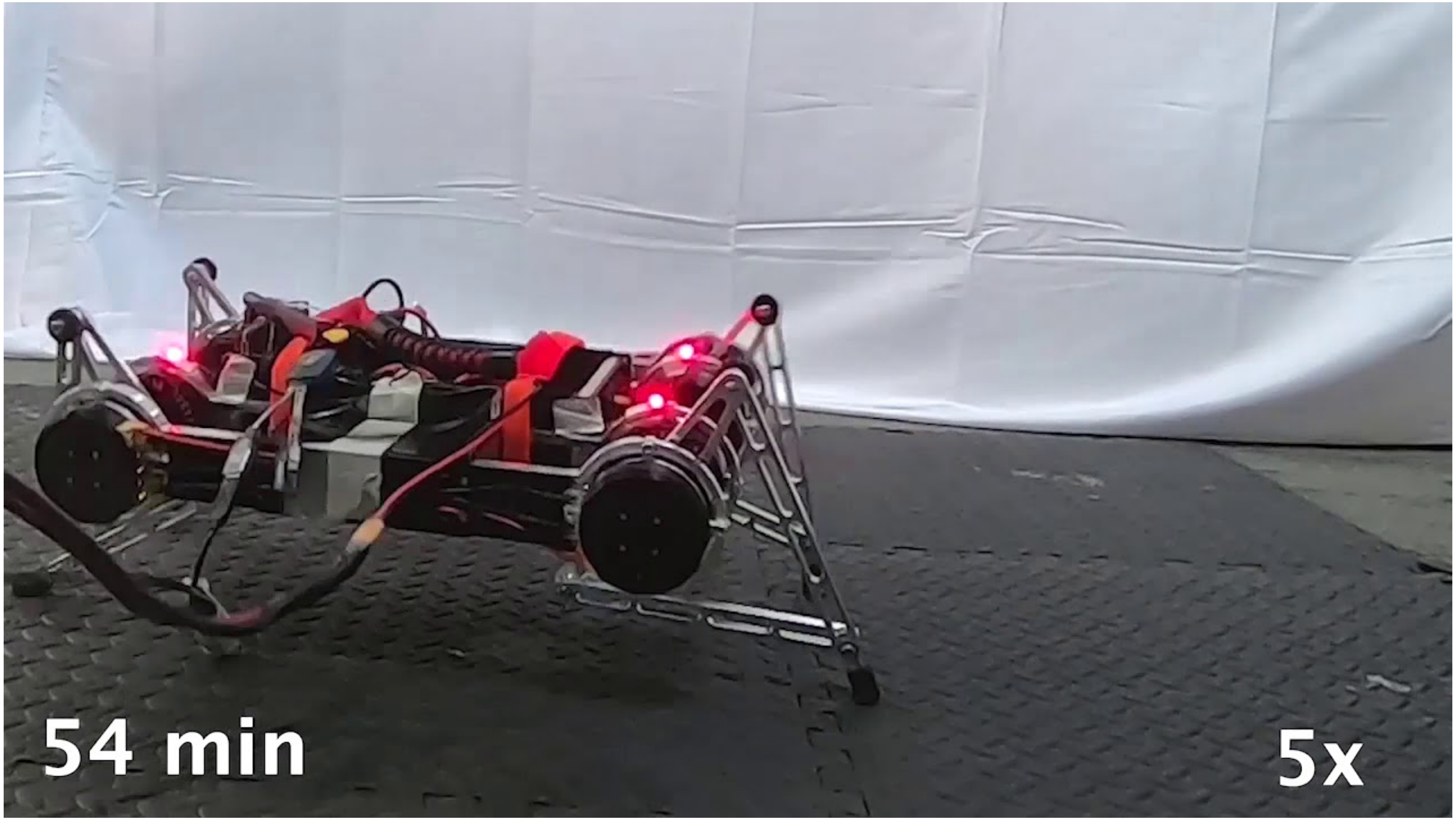
Tuomas Haarnoja   Aurick Zhou   Pieter Abbeel   Sergey Levine

- Why Soft Actor-Critic ?

—> sample efficient

—> very stable

—> exploration more efficient


54 min
5x

- Some general points:

- actor-critic

- off-policy algorithm

- continuous state and action spaces

Entropy-regularized RL setting

$$\mathrm{H}(X) = -\sum_{i=1}^{n} \mathrm{P}\left(x_i\right) \log \mathrm{P}\left(x_i\right) = -\mathbb{E}\left[\log P(X)\right]$$

$$\pi^*_{\text{old}} = \arg\max_{\pi} \underset{\tau\sim\pi}{\mathrm{E}} \left[\sum_{t=0}^{\infty} \gamma^t \left(R\left(s_t, a_t, s_{t+1}\right)\right)\right]$$

$$\pi^* = \arg\max_{\pi} \underbrace{\underset{\tau\sim\pi}{\mathrm{E}} \left[\sum_{t=0}^{\infty} \gamma^t \left(R\left(s_t, a_t, s_{t+1}\right) + \alpha H\left(\pi\left(\cdot \mid s_t\right)\right)\right)\right]}_{V^{\pi}(s_0)}$$

Entropy-regularized RL setting

$$V^{\pi}(s) = \underset{\tau \sim \pi}{\mathrm{E}} \left[\sum_{t=0}^{\infty} \gamma^t \Big(R\left(s_t, a_t, s_{t+1}\right) + \alpha H \Big(\pi \left(\cdot \mid s_t \right) \Big) \Big) \mid s_0 = s \right]$$

$$Q^{\pi}(s, a) = \underset{\tau \sim \pi}{\mathrm{E}} \left[\sum_{t=0}^{\infty} \gamma^t R\left(s_t, a_t, s_{t+1}\right) + \alpha \sum_{t=1}^{\infty} \gamma^t H \Big(\pi \left(\cdot \mid s_t \right) \Big) \mid s_0 = s, a_0 = a \right]$$

Equation de Bellman pour $Q^{\pi}(s, a)$:

$$Q^{\pi}(s, a) = \underset{\substack{s' \sim P \\ a' \sim \pi}}{\mathrm{E}} \left[R\left(s, a, s'\right) + \gamma V^{\pi}(s') \right]$$

$$Q^{\pi}(s, a) = \underset{\substack{s' \sim P \\ a' \sim \pi}}{\mathrm{E}} \left[R\left(s, a, s'\right) + \gamma \Big(Q^{\pi}\left(s', a'\right) + \alpha H \Big(\pi \left(\cdot \mid s' \right) \Big) \Big) \right]$$

Critic

Loss function of the Critic: MSBE (*Mean Squared Bellman Error*)

$$L(\phi, \mathscr{D}) = \underset{(s,a,r,s',d) \sim \mathscr{D}}{\mathrm{E}} \left[\left(Q_\phi(s, a) - Q^\pi(s, a) \right)^2 \right]$$

approximator

target

Critic

Loss function of the Critic: MSBE (*Mean Squared Bellman Error*)

$$L(\phi, \mathcal{D}) = \underset{(s,a,r,s',d)\sim\mathcal{D}}{\mathrm{E}}\left[\left(Q_\phi(s,a) - Q^\pi(s,a)\right)^2\right]$$

approximator

target

$$Q^\pi(s,a) \approx R\,(s,a,s') + \gamma\left(Q^\pi\,(s',\tilde{a}') - \alpha \log \pi\left(\tilde{a}' \mid s'\right)\right) \quad \text{avec} \quad \tilde{a}' \sim \pi_\theta\left(\cdot \mid s'\right)$$

Critic

Loss function of the Critic: MSBE (*Mean Squared Bellman Error*)

$$L(\phi, \mathscr{D}) = \underset{(s,a,r,s',d)\sim\mathscr{D}}{\mathrm{E}}\left[\left(Q_\phi(s,a) - y(r,s',d)\right)^2\right]$$

approximator

target

$$y(r,s',d) = R\,(s,a,s') + \gamma(1-d)\Big(Q_\phi\,(s',\tilde{a}') - \alpha \log \pi\,\big(\tilde{a}' \mid s'\big)\Big) \qquad \text{avec} \quad \tilde{a}' \sim \pi_\theta\big(\,\cdot \mid s'\big)$$

Critic

1st trick: Target Networks

Loss function of the Critic: MSBE (*Mean Squared Bellman Error*)

$$L(\phi, \mathcal{D}) = \underset{(s,a,r,s',d)\sim\mathcal{D}}{\mathrm{E}}\left[\left(\underbrace{Q_\phi(s,a)}_{\text{approximator}} - \underbrace{y(r,s',d)}_{\text{target}}\right)^2\right]$$

$$y(r,s',d) = R\left(s,a,s'\right) + \gamma(1-d)\left(Q_{\phi_{targ}}(s',\tilde{a}') - \alpha \log \pi\left(\tilde{a}' \mid s'\right)\right) \quad \text{avec} \quad \tilde{a}' \sim \pi_\theta\left(\cdot \mid s'\right)$$

$$\phi_{\text{targ}} \leftarrow \rho\phi_{\text{targ}} + (1-\rho)\phi$$

Silver, David, et al. "Deterministic policy gradient algorithms." *International conference on machine learning*. PMLR, 2014.

Critic

1st trick: Target Networks

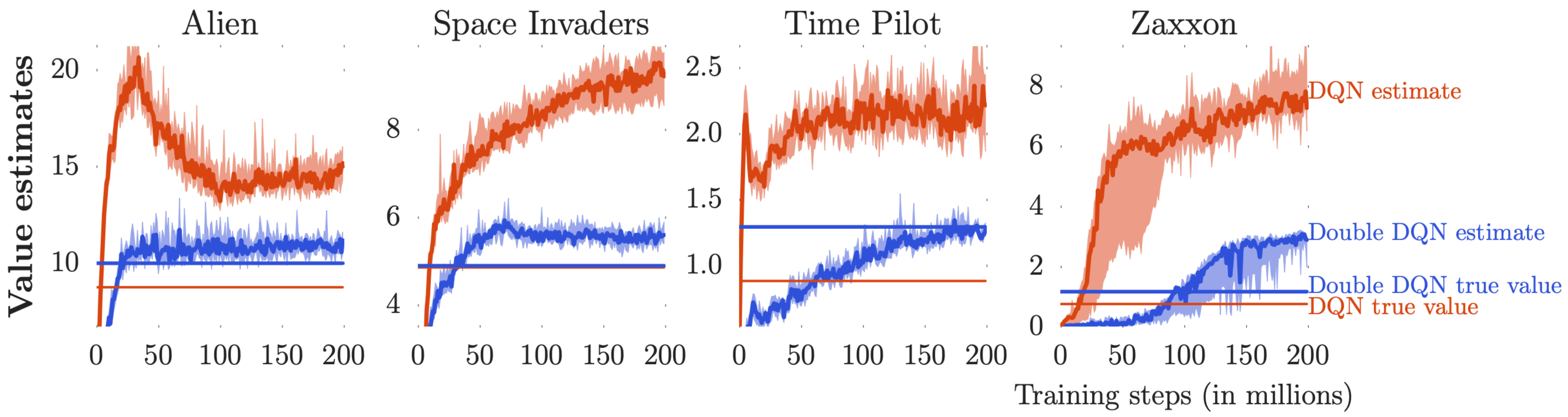
Loss function of the Critic: MSBE (*Mean Squared Bellman Error*)

$$L(\phi, \mathcal{D}) = \underset{(s,a,r,s',d)\sim\mathcal{D}}{\mathrm{E}} \left[\left(Q_\phi(s,a) - y(r,s',d) \right)^2 \right]$$

approximator   target

$$y(r,s',d) = R\,(s,a,s') + \gamma(1-d)\left(Q_{\phi_{targ}}(s',\tilde{a}') - \alpha \log \pi \left(\tilde{a}' \mid s' \right) \right) \quad \text{avec} \quad \tilde{a}' \sim \pi_\theta \left(\cdot \mid s' \right)$$

Critic



Source: Deep Reinforcement Learning with Double Q-learning (Hasselt et al., 2015)

Critic

1^{st} trick: Target Networks

Loss function of the Critic: MSBE (*Mean Squared Bellman Error*)

$$L(\phi, \mathcal{D}) = \underset{(s,a,r,s',d)\sim\mathcal{D}}{\mathrm{E}} \left[\left(Q_\phi(s,a) - y(r,s',d) \right)^2 \right]$$

approximator target

$$y(r,s',d) = R\,(s,a,s') + \gamma(1-d)\left(Q_{\phi_{targ}}(s',\tilde{a}') - \alpha \log \pi\left(\tilde{a}' \mid s'\right)\right) \quad \text{avec} \quad \tilde{a}' \sim \pi_\theta\left(\,\cdot \mid s'\right)$$

Critic

2^{nd} trick: Clipped Double Q-network —> ϕ_1 and ϕ_2

Loss function of the Critic: MSBE (*Mean Squared Bellman Error*)

For $i = 1,2$:

$$L(\phi_i, \mathcal{D}) = \underset{(s,a,r,s',d)\sim\mathcal{D}}{\mathrm{E}} \left[\left(Q_{\phi_i}(s,a) - y(r,s',d) \right)^2 \right]$$

approximator

target

$$y(r,s',d) = R\,(s,a,s') + \gamma(1-d)\left(\min_{j=1,2} Q_{\phi_{\text{targ},j}}(s',\tilde{a}') - \alpha \log \pi\left(\tilde{a}' \mid s'\right) \right) \quad \text{avec} \quad \tilde{a}' \sim \pi_\theta\left(\cdot \mid s' \right)$$

Actor

$$\pi^* = \arg\max_{\pi} V^{\pi}(s)$$

Objective function of the Actor for each state s:

$$V^{\pi}(s) = \mathop{\mathrm{E}}_{a \sim \pi} \left[Q^{\pi}(s, a) - \alpha \log \pi(a \mid s) \right]$$

Actor

$$\pi^* = \arg\max_{\pi} V^{\pi}(s)$$

Objective function of the Actor for each state s:

$$V^{\pi}(s) = \underset{\tilde{a} \sim \pi_\theta(\,\cdot\mid s)}{\mathrm{E}} \left[\min_{j=1,2} Q_{\phi_{\text{targ},j}}(s, \tilde{a}) - \alpha \log \pi_\theta(\tilde{a} \mid s) \right]$$

Actor

$\pi^* = \arg\max_{\pi} V^{\pi}(s)$

Objective function of the Actor for each state s:

$$V^{\pi}(s) = \mathop{\mathrm{E}}_{\tilde{a} \sim \pi_{\theta}(\,\cdot\,|\,s)} \left[\min_{j=1,2} Q_{\phi_{\mathrm{targ},j}}(s, \tilde{a}) - \alpha \log \pi_{\theta}(\tilde{a} \mid s) \right]$$

Pain point: the distribution depends on policy params

Actor

3^{rd} Trick : reparameterization

Objective function of the Actor for each state s:

$$V^{\pi}(s) = \mathop{\mathrm{E}}_{\xi \sim \mathcal{N}} \left[Q^{\pi_\theta}\left(s, \tilde{a}_\theta(s, \xi)\right) - \alpha \log \pi_\theta\left(\tilde{a}_\theta(s, \xi) \mid s\right) \right]$$

$$\tilde{a}_\theta(s, \xi) = \tanh\left(\mu_\theta(s) + \sigma_\theta(s) \odot \xi\right), \quad \xi \sim \mathcal{N}(0,I)$$

Actor

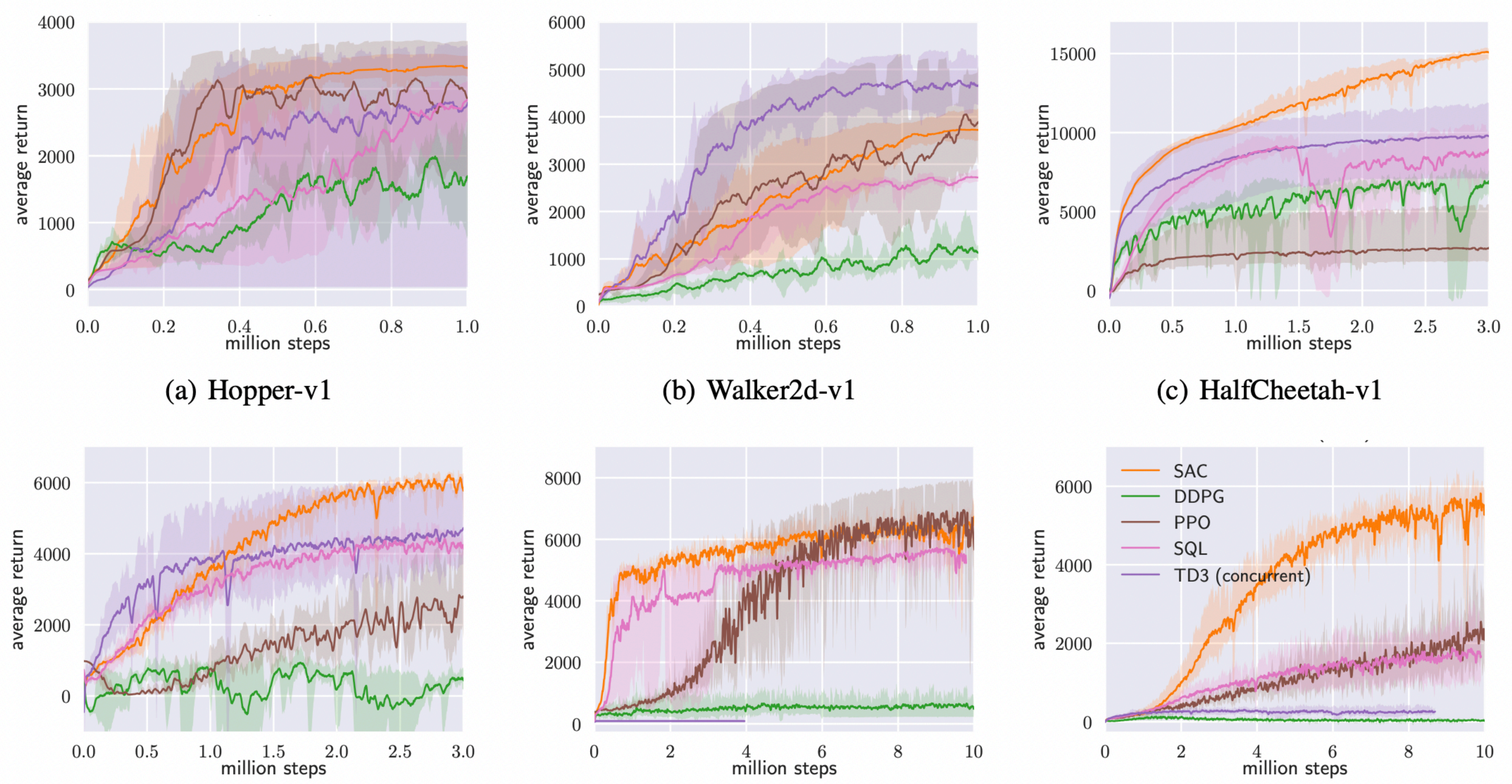
3^{rd} Trick : reparameterization

Objective function of the Actor:

$$\max_{\theta} \underset{\substack{s \sim \mathcal{D} \\ \xi \sim \mathcal{N}}}{\mathrm{E}} \left[\min_{j=1,2} Q_{\phi_j}\left(s, \tilde{a}_\theta(s,\xi)\right) - \alpha \log \pi_\theta\left(\tilde{a}_\theta(s,\xi) \mid s\right) \right]$$

$$\tilde{a}_\theta(s,\xi) = \tanh\left(\mu_\theta(s) + \sigma_\theta(s) \odot \xi\right), \quad \xi \sim \mathcal{N}(0,I)$$

Results



(a) Hopper-v1 (b) Walker2d-v1 (c) HalfCheetah-v1

(d) Ant-v1 (e) Humanoid-v1 (f) Humanoid (rllab)

References

- Haarnoja, Tuomas, et al. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor." *International conference on machine learning*. PMLR, 2018.

- Silver, David, et al. "Deterministic policy gradient algorithms." *International conference on machine learning*. PMLR, 2014.

- Fujimoto, Scott, Herke Hoof, and David Meger. "Addressing function approximation error in actor-critic methods." *International conference on machine learning*. PMLR, 2018.

- https://spinningup.openai.com/en/latest/algorithms/sac.html