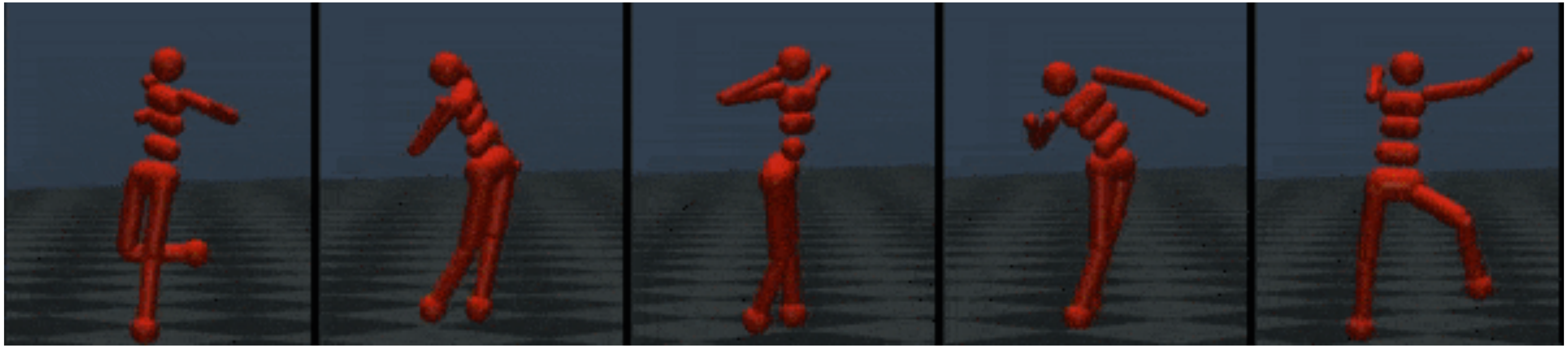
Reinforcement Learning

HSE, winter - spring 2025

Lecture 6: Continuous Control

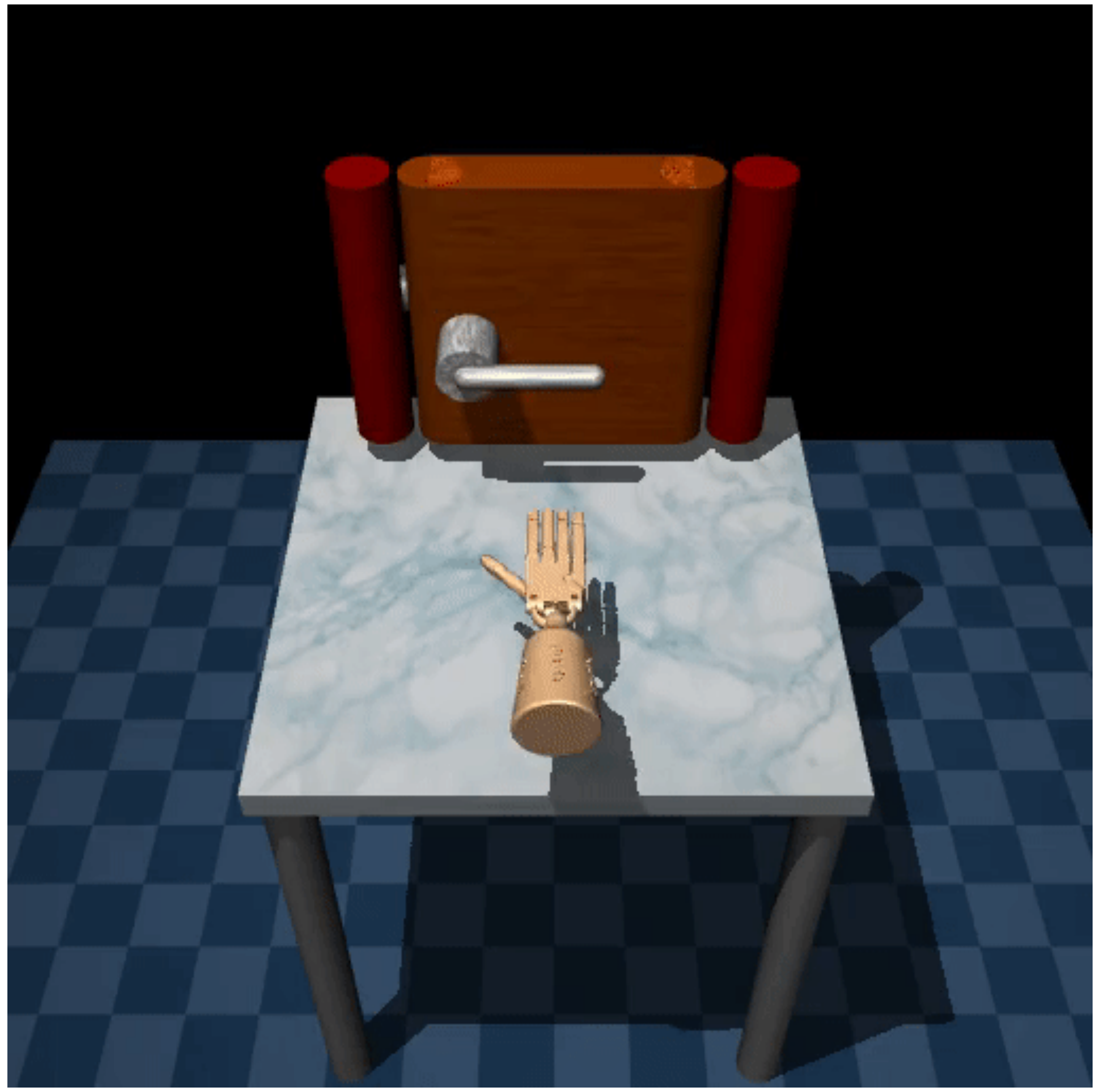


Source

Sergei Laktionov
slaktionov@hse.ru
LinkedIn

Continuous Control Tasks

- Action space $\mathscr{A} = [-1,1]^A$
- Dense reward



Source

Recap: Value-based vs Policy-based

- Value-based (DQN):

  1. Policy evaluation:

     Learn Q^* using Bellman target $r + \gamma \max_{a'} Q_\phi(s', a')$

  2. Policy improvement:

     Recover policy greedily w.r.t. $Q_\phi(s, a)$

- Policy-based (REINFORCE, A2C, PPO):

  1. Policy evaluation:

     Learn critic V_ϕ to estimate the quality of the current policy

  2. Policy improvement:

     Learn policy π_θ directly calculating the gradient using log-derivative trick of $J(\theta)$ w.r.t. policy parameters θ

Recap: Value-based vs Policy-based

- Value-based (DQN):
  - Only applicable to the discrete action space due to $argmax_a Q(s, a)$
  - Artificial exploration with ε-greedy policies
  - Off-policy algorithm, high sample efficiency thanks to the replay buffer.
  - 1-step target, low signal propagation
- Policy-based (REINFORCE, A2C, PPO):
  - Applicable to both discrete and continuous action spaces
  - Natural exploration with stochastic policies
  - On-policy, lower sample efficiency, Replay Buffer can not be used
  - N-step target, GAE

Policy Improvement as Optimisation

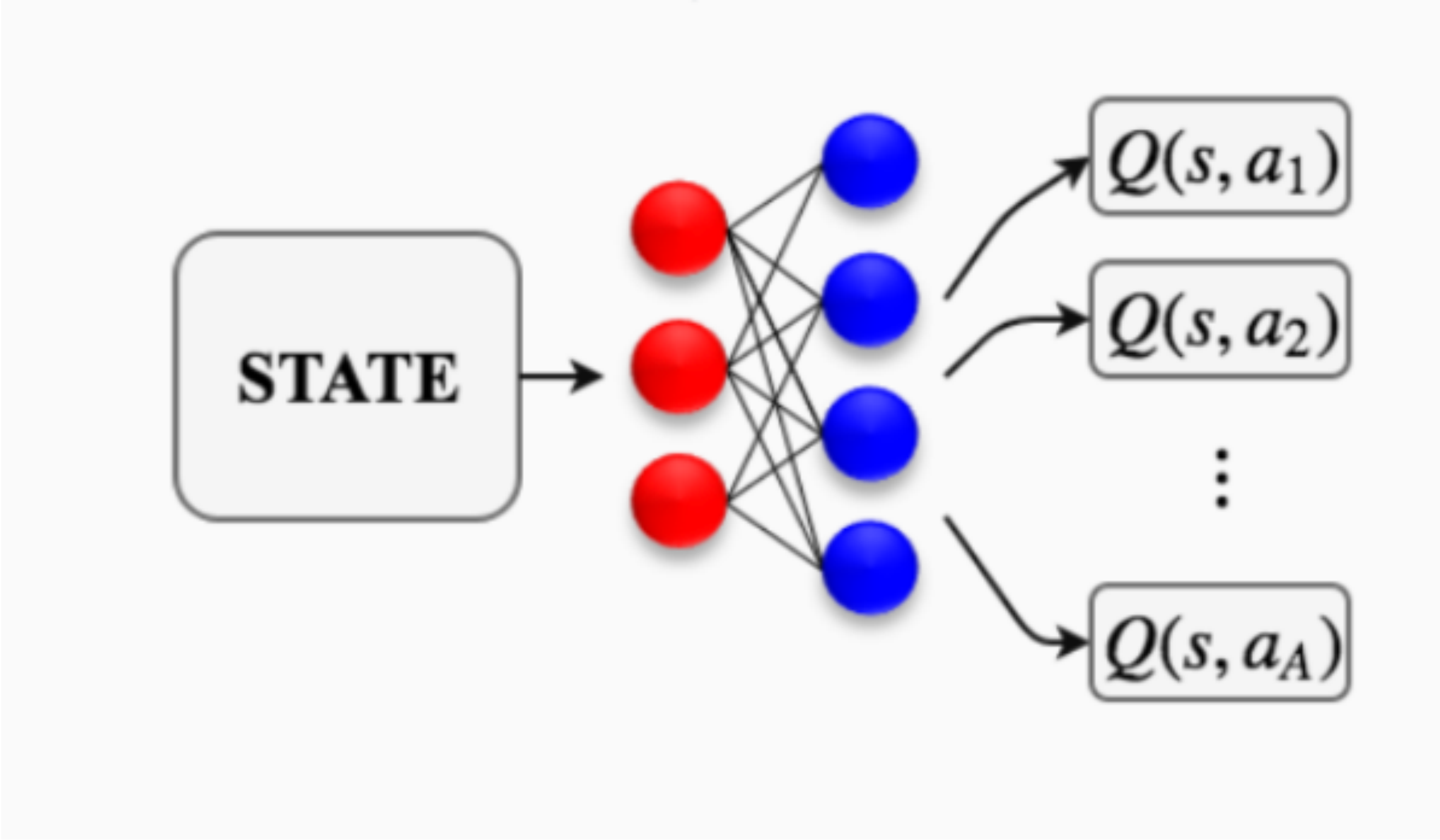
- Recap: If $\mathbb{E}_{a\sim\pi}Q^{\pi_{old}}(s,a) \geq V^{\pi_{old}}(s)$ for all s then π is not worse than π_{old}

Policy Improvement as Optimisation

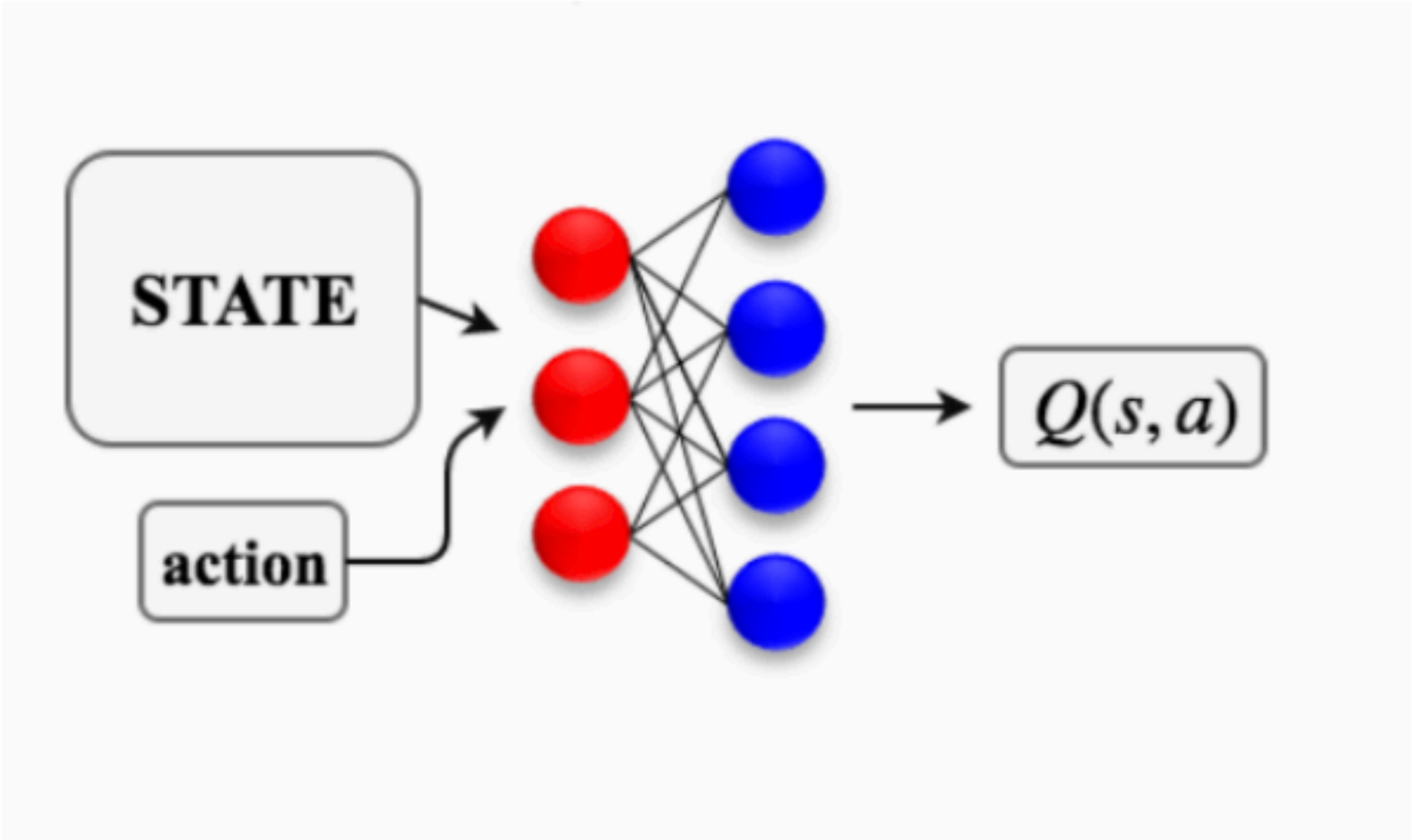
- Recap: If $\mathbb{E}_{a\sim\pi}Q^{\pi_{old}}(s,a) \geq V^{\pi_{old}}(s)$ for all s then π is not worse than π_{old}
- For dicrete actions we can simply take $\pi(s) = argmax_a Q^{\pi_{old}}(s,a)$
- General optimisation: $\mathbb{E}_{s\sim\rho}\mathbb{E}_{a\sim\pi}Q^{\pi_{old}}(s,a) \to \max\limits_{\pi}$

Policy Improvement as Optimisation

Source

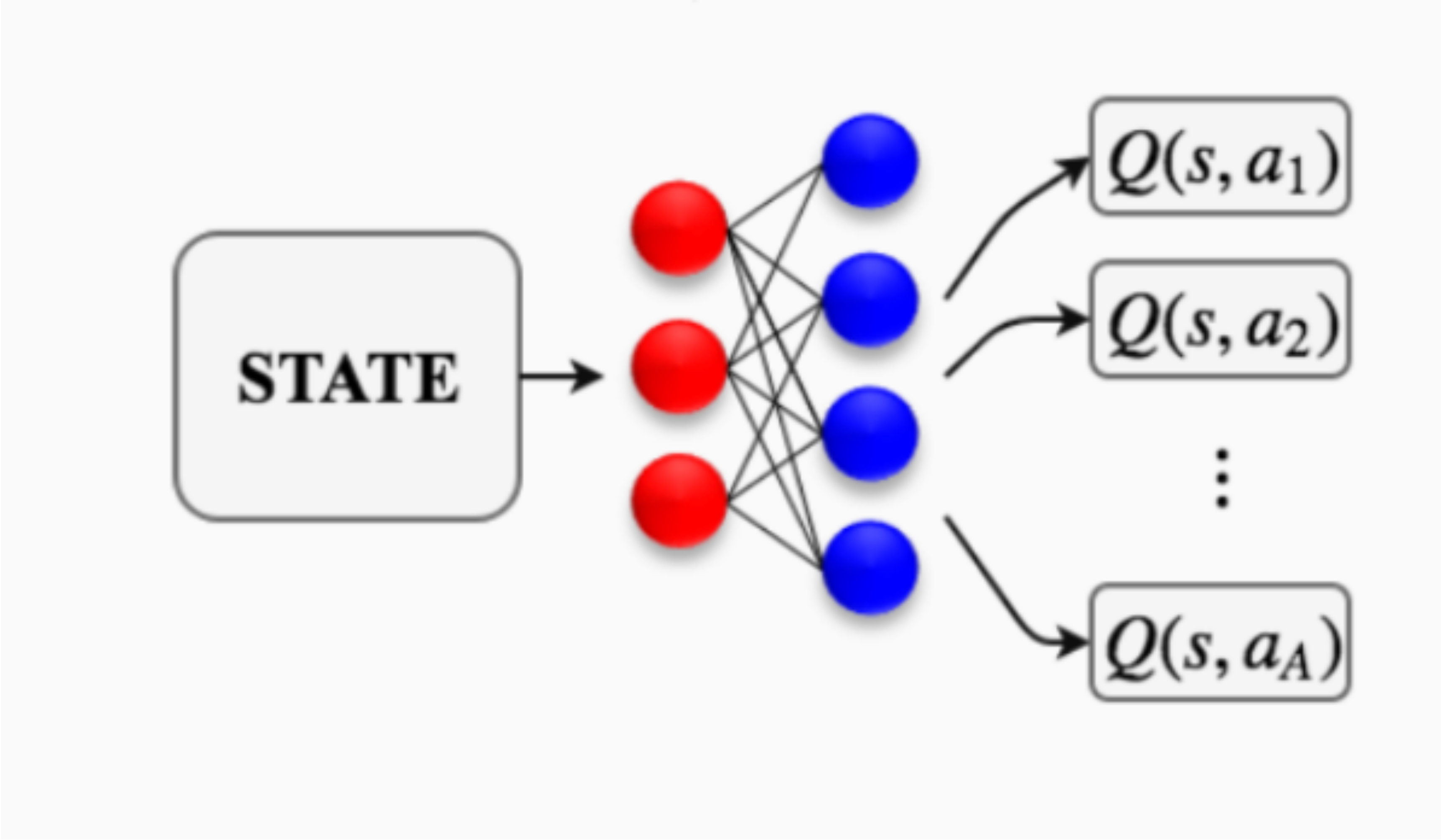


Source



Policy Improvement as Optimisation

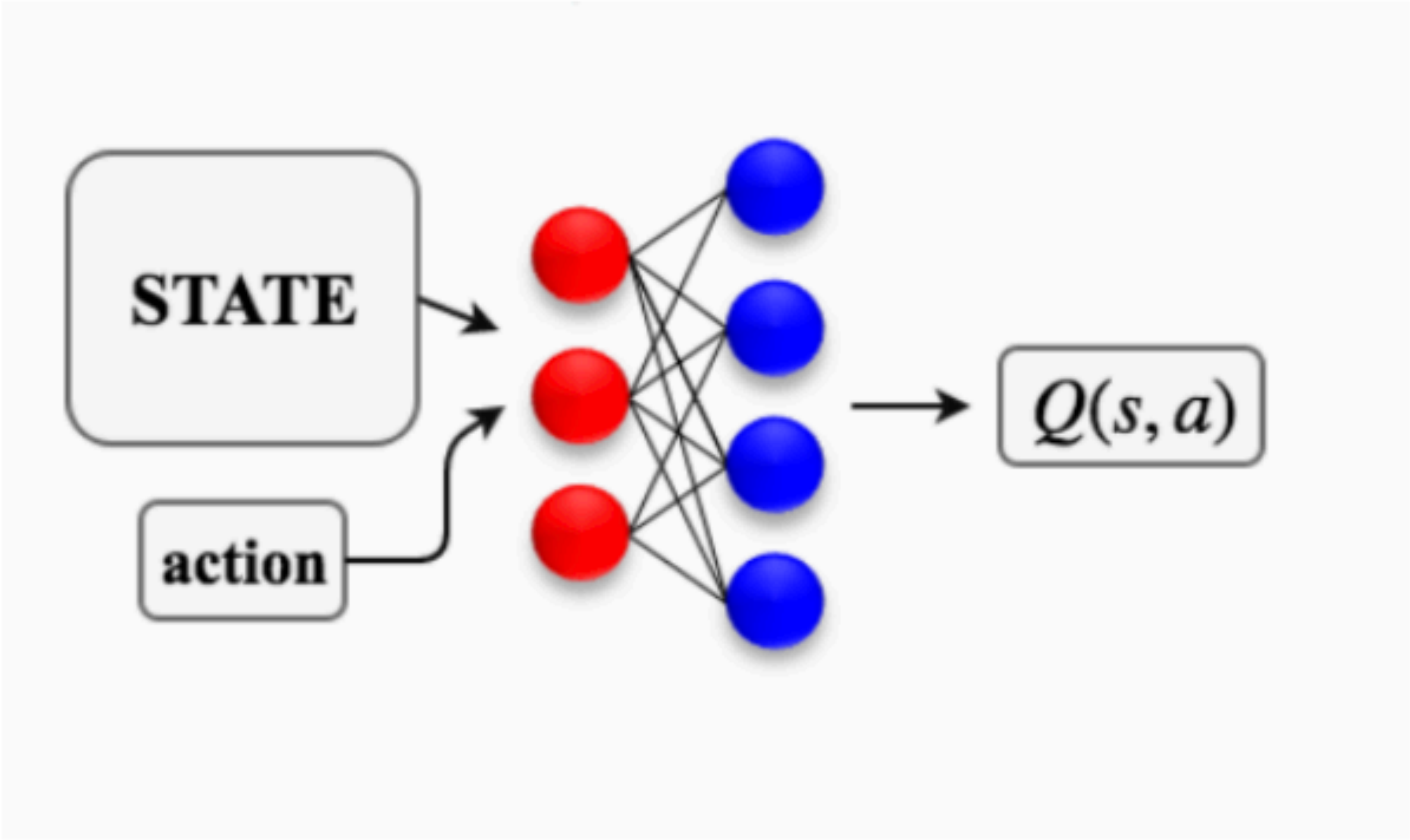
Source



$$Q_\phi(s, a) \to \max_a$$

DQN

Source



$$Q_\phi(s, \mu_\theta(s)) \to \max_\theta$$

$\mu_\theta(s)$ is a deterministic parametrised policy

DDPG

Exploration

An advantage of off-policy algorithms is that we can treat the problem of exploration independently from the learning algorithm.

$a = \mu_\theta(s) + \varepsilon$, where:

1. Gaussian noise: $\varepsilon \sim \mathcal{N}(0,\sigma^2)$

2. Ornstein-Uhlenbeck process: $\varepsilon_t = \alpha\varepsilon_{t-1} + \nu, \nu \sim \mathcal{N}(0,\sigma^2)$

Deep Deterministic Policy Gradient

Actor $\pi_\theta(s)$, critic $Q_\phi(s, a)$, target actor $\pi_{\theta^-}(s)$, target critic $Q_{\phi^-}(s, a)$.

On each step:

- Observe s, choose $a = \pi_\theta(s) + \varepsilon$, get $s', r, done$, put the transition into the buffer
- On the batch of transitions $(s_i, a, r_i, s_i', done_i)_{i=1}^B$, sampled from the replay buffer, perform:

    1. Policy evaluation: $\frac{1}{B}\sum_{i=1}^{B}(y_i - Q_\phi(s_i, a_i))^2 \to \min_\phi$, where $y_i = r_i + \gamma(1 - done_i)Q_{\phi^-}(s_i', \pi_{\theta^-}(s_i'))$
    2. Policy improvement: $\frac{1}{B}\sum_{i=1}^{B} Q_\phi(s_i, \pi_\theta(s_i)) \to \max_\theta$

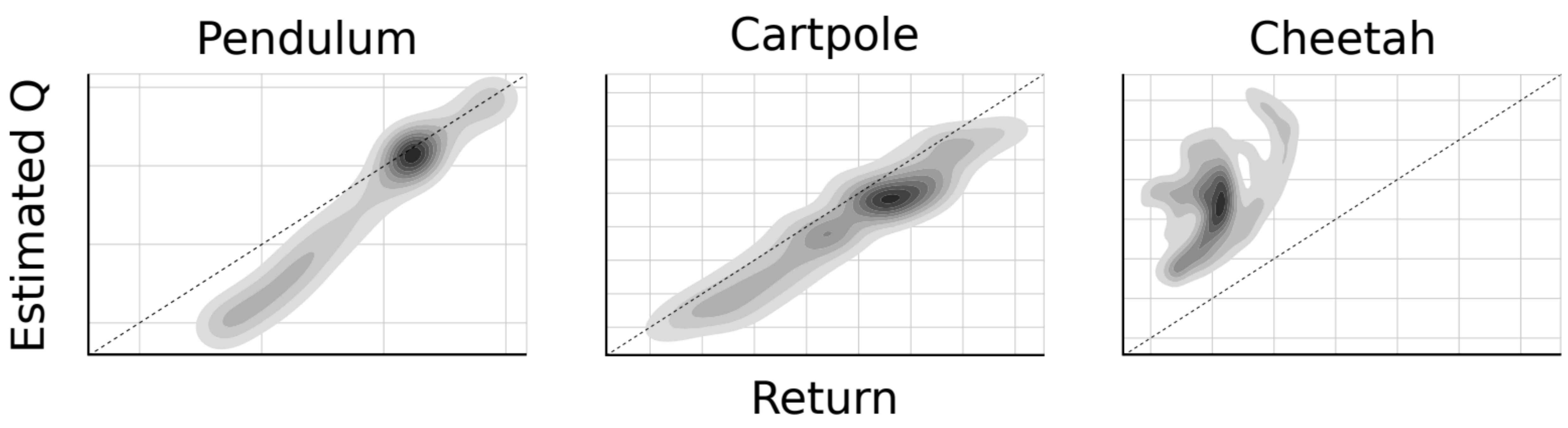
- Soft-update the actor and critic:
    - $\theta^- = \tau\theta + (1-\tau)\theta^-, \phi^- = \phi\theta + (1-\tau)\phi^-$

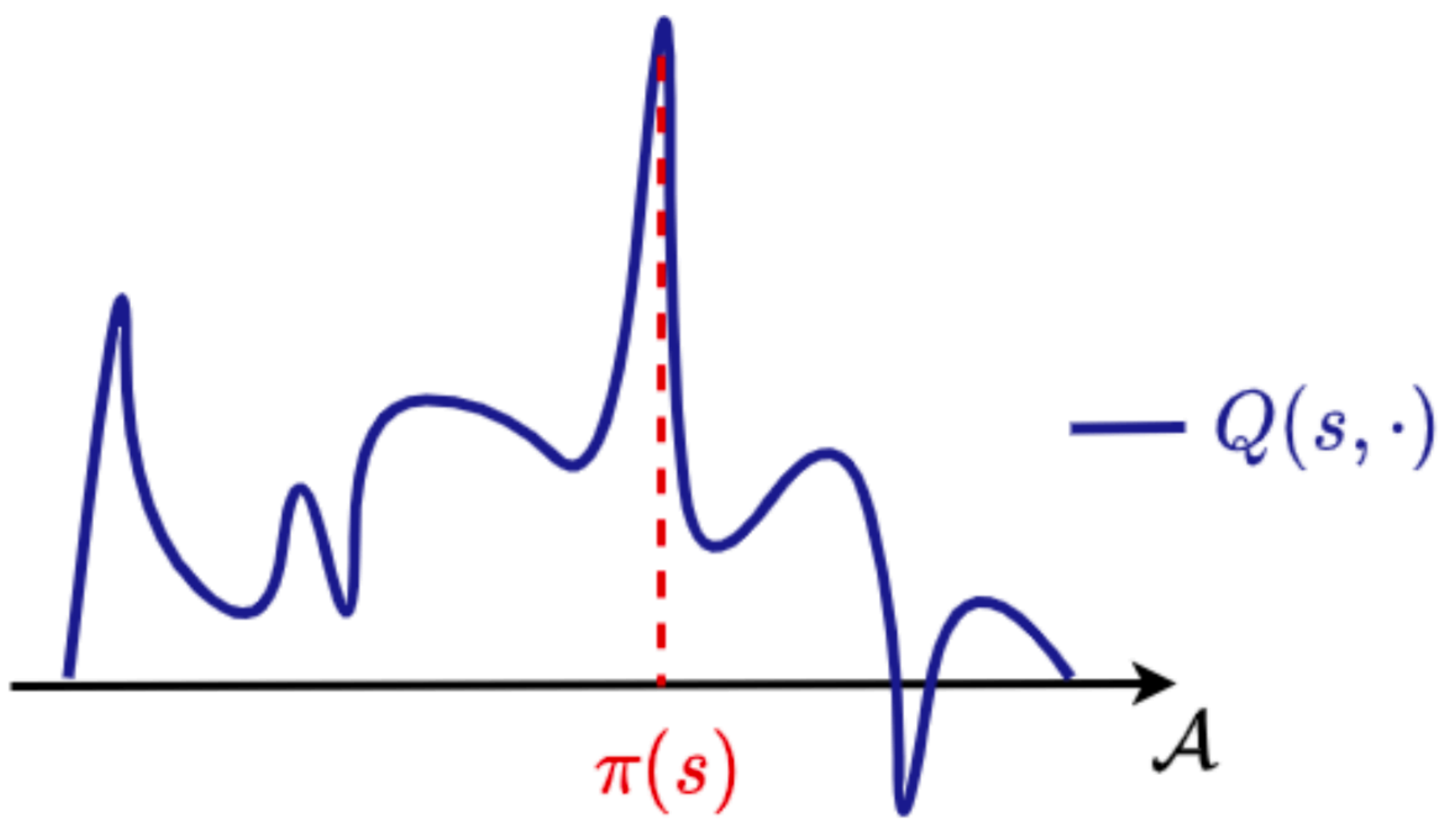
Issues

1. Overestimation bias

2. Sharpe peaks

3. Volatility that normally arises in DDPG because of how a policy update changes the target.





Twin Delayed DDPG

Clipped Double-Q Learning:

$$y = r + \gamma \min_{i=1,2} Q_{\phi_i^-}(s', \mu_{\theta^-}(s'))$$

Delayed Policy Updates: Update the policy (and target networks) less frequently than the Q-function.

Target Policy Smoothing: Add noise to the target action, to make it harder for the policy to exploit Q-function errors:

$$y = r + \gamma \min_{i=1,2} Q_{\phi_i^-}(s', a'),\ a' = \mu_{\theta^-}(s) + \varepsilon',$$

$$\varepsilon' \sim clip(\mathcal{N}(0, \sigma^2 I), -c, c)$$

Deterministic Policy Gradient

For deterministic policy $\pi_\theta : \mathcal{S} \rightarrow \mathcal{A}$:

$$\nabla_\theta J(\theta) = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_{\pi_\theta}} \nabla_\theta \pi_\theta \nabla_a Q^{\pi_\theta}(s, a) \,|_{a=\pi_\theta(s)}$$

Deterministic Policy Gradient

For deterministic policy $\pi_\theta : \mathcal{S} \to \mathcal{A}$:

$$\nabla_\theta J(\theta) = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_{\pi_\theta}} \nabla_\theta \pi_\theta \nabla_a Q^{\pi_\theta}(s, a) \,|_{a=\pi_\theta(s)}$$

Surrogate objective for policy gradient:

$$L_{\pi_{old}}(\theta) = \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_{\pi_{old}}} [Q^{\pi_{old}}(s, \pi_\theta(s))]$$

Surrogate objective for any policy improvement:

$$L_{\pi_{old}}(\theta) = \frac{1}{1-\gamma} \mathbb{E}_{s \sim \rho} [Q^{\pi_{old}}(s, \pi_\theta(s))]$$

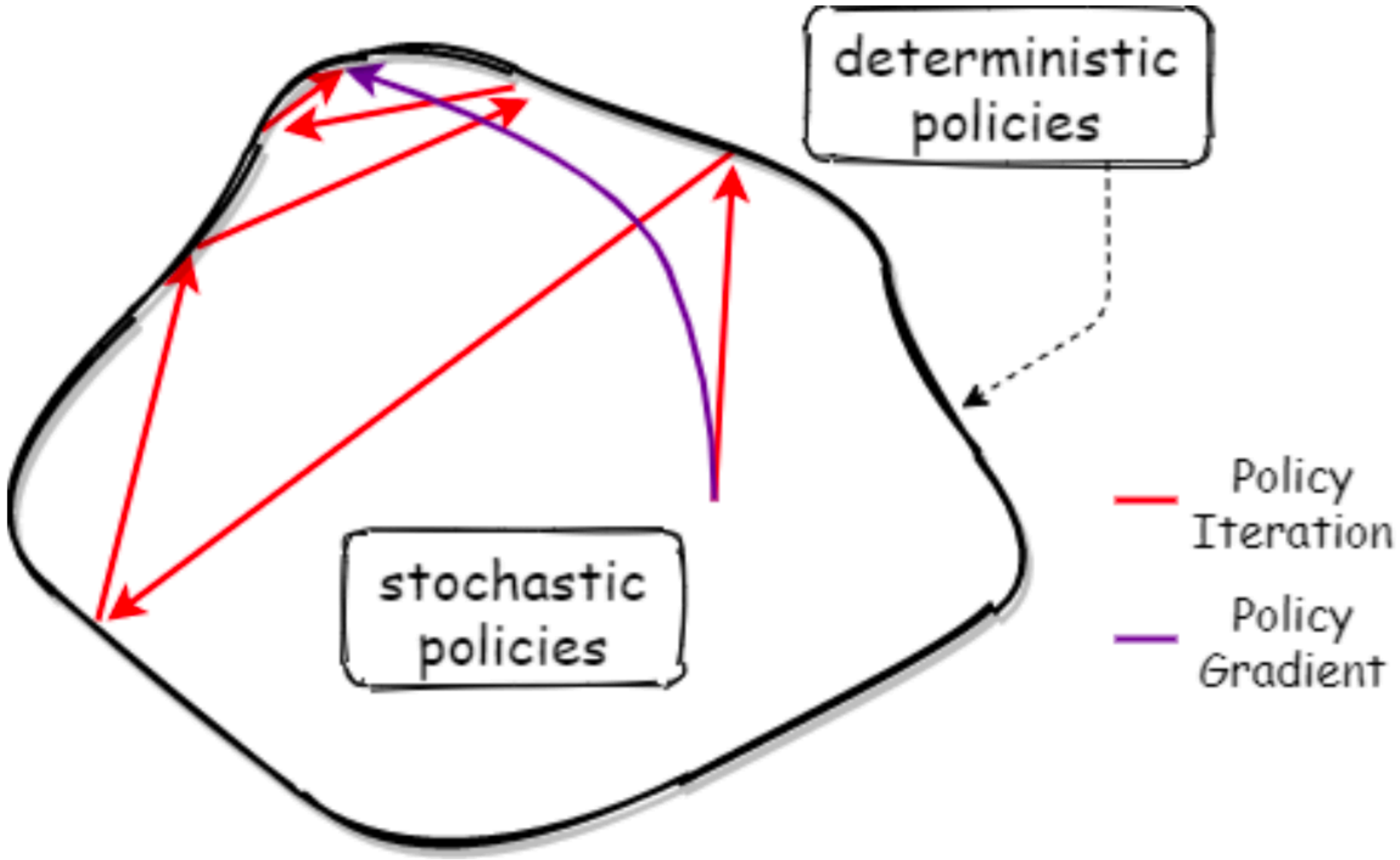
Deterministic Policy Gradient

Surrogate objective for deterministic policy gradient:

$$L_{\pi_{old}}(\theta) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{\pi_{old}}}[Q^{\pi_{old}}(s, \pi_\theta(s))]$$

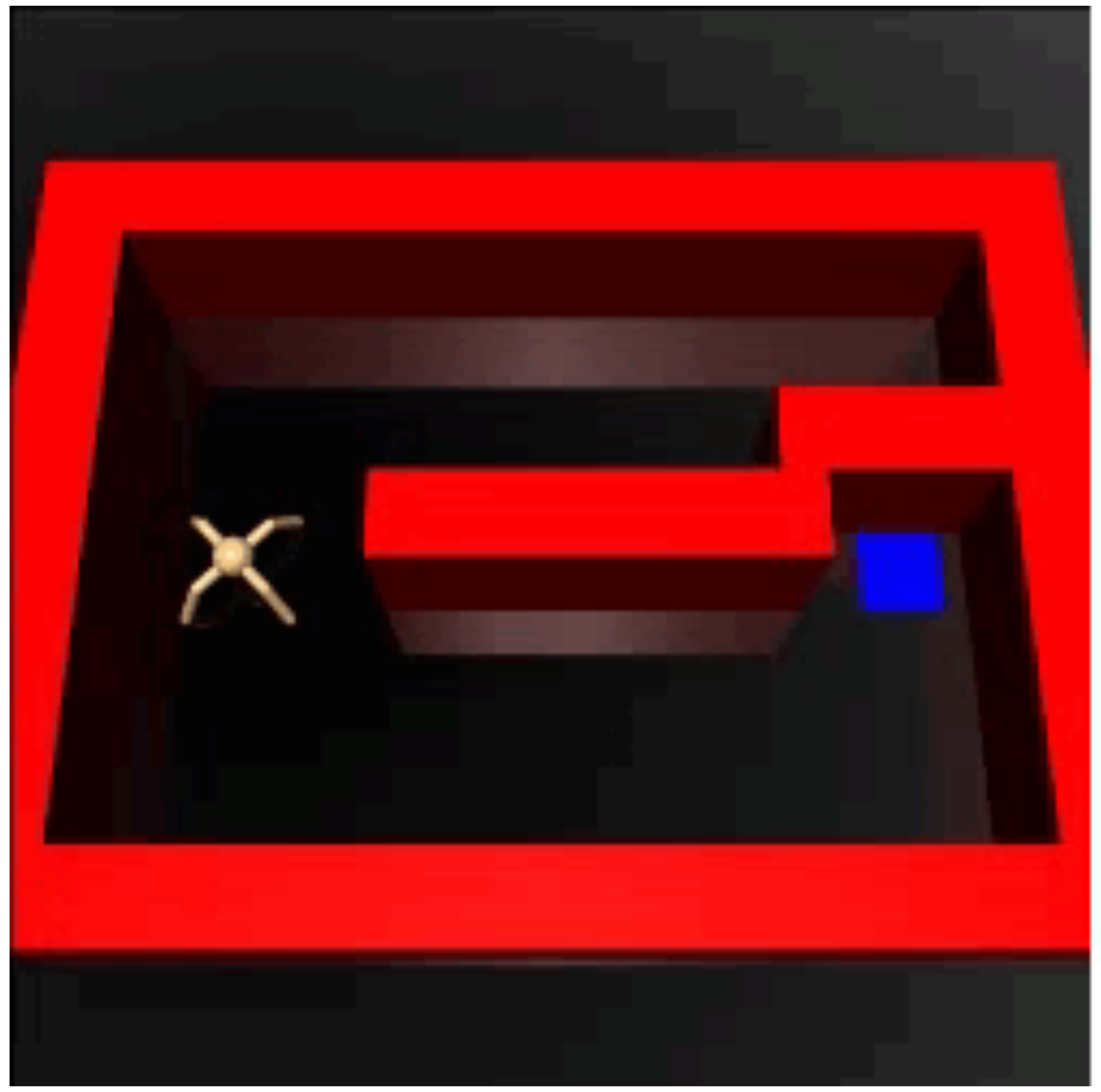
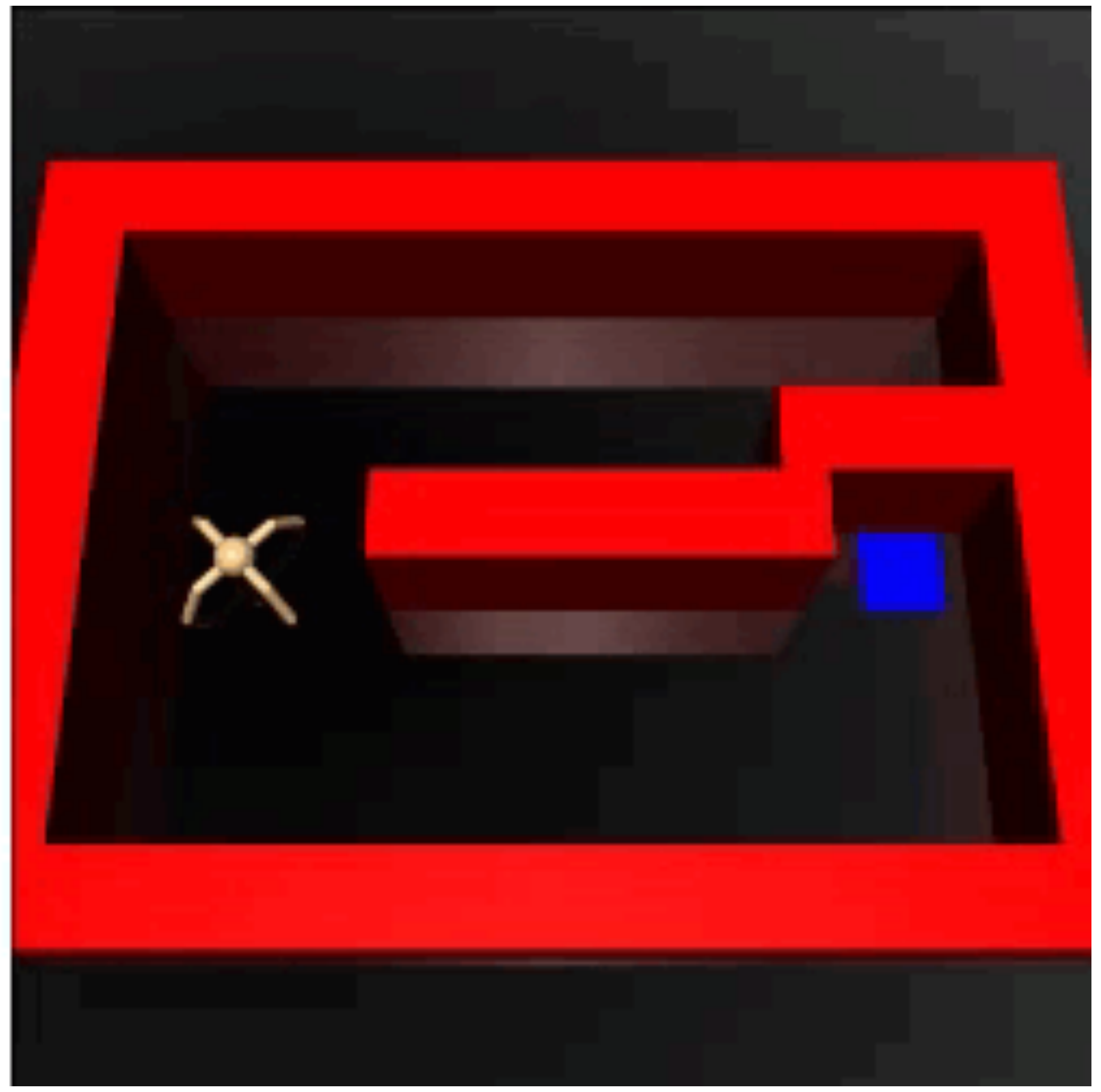
Surrogate objective for any policy improvement:

$$L_{\pi_{old}}(\theta) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim \rho}[Q^{\pi_{old}}(s, \pi_\theta(s))]$$



Stochastic Policies

- So far, we’ve introduced two off-policy algorithms learning only deterministic policies, where exploration is artificially maintained by adding noise.
- We want to train stochastic policies to have natural exploration, which prevents our agents from getting stuck in local optima.
- We also want to prevent our stochastic policies from becoming “too deterministic” very quickly.

Source

Policy Gradient

$$\mathbb{E}_s \mathbb{E}_{a \sim \pi_\theta(.|s)}[Q^{\pi_{old}}(s, a)] \to \max_\theta$$

Let's take $\mathbb{E}_s \nabla_\theta \mathbb{E}_{a \sim \pi_\theta(.|s)}[Q^{\pi_{old}}(s, a)]$:

REINFORCE

$$\mathbb{E}_s \mathbb{E}_{a \sim \pi_\theta(.|s)}[\nabla_\theta \log \pi_\theta(a \mid s) Q^{\pi_{old}}(s, a)]$$

Reparametrisation Trick

$$\mathbb{E}_s \mathbb{E}_{\varepsilon \sim p(.)}[\nabla_\theta Q^{\pi_{old}}(s, f_\theta(s, \varepsilon))]$$

If $a \sim \pi(\,.\mid s)$ is equivalent to $a = f_\theta(s, \varepsilon)$,
Where f_θ is a deterministic function,
$\varepsilon \sim p(\,.\,)$ is a non-parametric distribution.

Policy Gradient

REINFORCE

$$\mathbb{E}_s \mathbb{E}_{a \sim \pi_\theta(.|s)}[\nabla_\theta \log \pi_\theta(a \,|\, s) Q^{\pi_{old}}(s, a)]$$

1. Softmax policy: $a \sim softmax(logit_\theta(s))$

2. Deterministic policy: $a = \pi_\theta(s)$

3. Gaussian policy: $a \sim \mathcal{N}(\mu_\theta(s), \sigma_\theta^2(s) I)$

4. Mixture of gaussian: $a \sim \sum_{i=1}^{K} w_\theta^i(s) \mathcal{N}(\mu_\theta^i(s), (\sigma_\theta^i(s))^2 I)$

Reparametrisation Trick

$$\mathbb{E}_s \mathbb{E}_{\varepsilon \sim p(.)}[\nabla_\theta Q^{\pi_{old}}(s, f_\theta(s, \varepsilon))]$$

If $a \sim \pi(\,.\,|\,s)$ is equivalent to $a = f_\theta(s, \varepsilon)$,
where f_θ is a deterministic function,
$\varepsilon \sim p(\,.\,)$ is a non-parametric distribution.

REINFORCE vs Reparametrisation Trick

Properties	REINFORCE	Reparameterization Trick
Differentiability requirements	Can work with Q(s, a) non-differentiable w.r.t. a	Needs a differentiable Q(s, a) w.r.t. a
Gradient variance	High variance; needs variance reduction techniques	Low variance due to implicit modeling of dependencies
Type of distribution	Works for both discrete and continuous action space	In the current form, only valid for continuous action spaces
Family of distribution	Works for a large class of distributions of x	It should be possible to reparameterize x as done above

Maximum Entropy RL

$$J_{soft}(\pi) = \mathbb{E}_{\tau\sim\pi}\sum_{t=0}^{\infty}\gamma^t[r_t + \alpha H(\pi(\,.\,|\,s_t))]$$, where $H(\pi(\,.\,|\,s))$ is an entropy.

Equivalent form:

$$J_{soft}(\pi) = \mathbb{E}_{\tau\sim\pi}\sum_{t=0}^{\infty}\gamma^t[r_t - \alpha\log\pi(a_t\,|\,s_t)]$$

Maximum Entropy RL

To eliminate the reward's dependency on current policy let's fix the following order:

$$s \longrightarrow H(\pi(\,.\,|\,s)) \longrightarrow a \longrightarrow r \longrightarrow s' \longrightarrow H(\pi(\,.\,|\,s')) \longrightarrow a' \longrightarrow \dots$$

Maximum Entropy RL

To eliminate the reward's dependency on current policy let's fix the following order:

$$s \longrightarrow H(\pi(\,.\,|\,s)) \longrightarrow a \longrightarrow r \longrightarrow s' \longrightarrow H(\pi(\,.\,|\,s')) \longrightarrow a' \longrightarrow \dots$$

$$V^{\pi}_{soft}(s) = \mathbb{E}_a[r(s,a) + \gamma \mathbb{E}_{s'} V^{\pi}_{soft}(s') - \alpha \log \pi(a\,|\,s)]$$

Maximum Entropy RL

To eliminate the reward's dependency on current policy let's fix the following order:

$$s \longrightarrow H(\pi(\,.\,|\,s)) \longrightarrow a \longrightarrow r \longrightarrow s' \longrightarrow H(\pi(\,.\,|\,s')) \longrightarrow a' \longrightarrow \dots$$

$$V^{\pi}_{soft}(s) = \mathbb{E}_a[r(s,a) + \gamma \mathbb{E}_{s'} V^{\pi}_{soft}(s') - \alpha \log \pi(a\,|\,s)]$$

$$Q^{\pi}_{soft}(s,a) = r(s,a) + \gamma \mathbb{E}_{s'} V^{\pi}_{soft}(s')$$

Maximum Entropy RL

To eliminate the reward's dependency on current policy let's fix the following order:

$$s \longrightarrow H(\pi(\,.\,|\,s)) \longrightarrow a \longrightarrow r \longrightarrow s' \longrightarrow H(\pi(\,.\,|\,s')) \longrightarrow a' \longrightarrow \dots$$

$$V^{\pi}_{soft}(s) = \mathbb{E}_a[r(s,a) + \gamma \mathbb{E}_{s'} V^{\pi}_{soft}(s') - \alpha \log \pi(a\,|\,s)]$$

$$Q^{\pi}_{soft}(s,a) = r(s,a) + \gamma \mathbb{E}_{s'} V^{\pi}_{soft}(s')$$

$$V^{\pi}_{soft}(s) = \mathbb{E}_a[Q^{\pi}_{soft}(s,a) - \alpha \log \pi(a\,|\,s)]$$

Maximum Entropy RL

To eliminate the reward's dependency on current policy let's fix the following order:

$$s \longrightarrow H(\pi(\,.\,|\,s)) \longrightarrow a \longrightarrow r \longrightarrow s' \longrightarrow H(\pi(\,.\,|\,s')) \longrightarrow a' \longrightarrow \dots$$

$$V^{\pi}_{soft}(s) = \mathbb{E}_a[r(s,a) + \gamma \mathbb{E}_{s'} V^{\pi}_{soft}(s') - \alpha \log \pi(a\,|\,s)]$$

$$Q^{\pi}_{soft}(s,a) = r(s,a) + \gamma \mathbb{E}_{s'} V^{\pi}_{soft}(s')$$

$$V^{\pi}_{soft}(s) = \mathbb{E}_a[Q^{\pi}_{soft}(s,a) - \alpha \log \pi(a\,|\,s)]$$

$$Q^{\pi}_{soft}(s,a) = r(s,a) + \gamma \mathbb{E}_{s'} \mathbb{E}_{a'}[Q^{\pi}_{soft}(s',a') - \alpha \log \pi(a'\,|\,s')]$$

Soft Policy Evaluation

For transition, (s, a, r, s') define a critic's target:

$$y_Q = r(s, a) + \gamma \mathbb{E}_{\boxed{a' \sim \pi(.|s')}}[Q_\phi(s', a') - \alpha \log \pi(a' \mid s')]$$

In general intractable

Soft Policy Evaluation

For transition, (s, a, r, s') define a critic's target:

$$y_Q = r(s, a) + \gamma \mathbb{E}_{\boxed{a' \sim \pi(.|s')}}[Q_\phi(s', a') - \alpha \log \pi(a' \mid s')]$$

In general intractable

- We can estimate the expectation using a sample from the policy

- Can learn V_ψ to approximate the expectation:

$$y_V = Q_\phi(s, a_\pi) - \alpha \log \pi(a_\pi \mid s), a_\pi \sim \pi(. \mid s)$$

$$y_Q = r(s, a) + \gamma V_\psi(s')$$

Policy Improvement

Policy Improvement in traditional RL

- If $\mathbb{E}_{a \sim \pi} Q^{\pi_{old}}(s, a) \geq V^{\pi_{old}}(s)$
  then π is not worse than π_{old}
- Optimisation:

$$\mathbb{E}_s \mathbb{E}_{a \sim \pi} Q^{\pi_{old}}(s, a) \to \max_{\pi}$$

- $\pi(s) = argmax_a Q^{\pi_{old}}(s, a)$

Policy Improvement in Max Entropy RL

Policy Improvement

Policy Improvement in traditional RL

- If $\mathbb{E}_{a\sim\pi} Q^{\pi_{old}}(s,a) \geq V^{\pi_{old}}(s)$ then π is not worse than π_{old}
- Optimisation:

$$\mathbb{E}_s \mathbb{E}_{a\sim\pi} Q^{\pi_{old}}(s,a) \to \max_\pi$$

- $\pi(s) = argmax_a Q^{\pi_{old}}(s,a)$

Policy Improvement in Max Entropy RL

- If $\mathbb{E}_{a\sim\pi}[Q^{\pi_{old}}_{soft}(s,a) - \alpha \log \pi(\,.\,|\,s)] \geq V^{\pi_{old}}_{soft}(s)$ then π is not worse than π_{old}
- Optimisation:

$$\mathbb{E}_{a\sim\pi}[Q^{\pi_{old}}_{soft}(s,a) - \alpha \log \pi(\,.\,|\,s)] \to \max_\pi$$

Policy Improvement

Policy Improvement in traditional RL

- If $\mathbb{E}_{a\sim\pi} Q^{\pi_{old}}(s,a) \geq V^{\pi_{old}}(s)$ then π is not worse than π_{old}
- Optimisation:

$$\mathbb{E}_s \mathbb{E}_{a\sim\pi} Q^{\pi_{old}}(s,a) \rightarrow \max_{\pi}$$

- $\pi(s) = argmax_a Q^{\pi_{old}}(s,a)$

Policy Improvement in Max Entropy RL

- If $\mathbb{E}_{a\sim\pi}[Q^{\pi_{old}}_{soft}(s,a) - \alpha \log \pi(\,.\,|\,s)] \geq V^{\pi_{old}}_{soft}(s)$ then π is not worse than π_{old}
- Optimisation:

$$\mathbb{E}_{a\sim\pi}[Q^{\pi_{old}}_{soft}(s,a) - \alpha \log \pi(\,.\,|\,s)] \rightarrow \max_{\pi}$$

- $\pi(a\,|\,s) \propto \exp\left(\frac{Q^{\pi_{old}}(s,a)}{\alpha}\right)$

Soft Policy Improvement

Actor π_θ learning:

$$\mathbb{E}_s[\mathbb{E}_{a\sim\pi_\theta} Q_\phi(s,a) - \alpha \log \pi_\theta(\,.\,|\,s)] \to \max_\theta$$

Soft Policy Improvement

Actor π_θ learning:

$$\mathbb{E}_s[\mathbb{E}_{a\sim\pi_\theta} Q_\phi(s,a) - \alpha \log \pi_\theta(\,.\,|\,s)] \to \max_\theta$$

Example:

- $\pi_\theta(\,.\,|\,s) = \mathcal{N}(\mu_\theta(s), \sigma^2_\theta(s)I)$
- $a \sim \pi_\theta(\,.\,|\,s) \iff a = \mu_\theta(s) + \sigma_\theta(s)\varepsilon, \varepsilon \sim \mathcal{N}(0,I)$
- $H(\pi_\theta(\,.\,|\,s))] = \sum_{i=1}^{A} \log \sigma^i_\theta(s)$
- $\mathbb{E}_s[\mathbb{E}_{\varepsilon\sim\mathcal{N}(0,I)} Q_\phi(s, \mu_\theta(s) + \sigma_\theta(s)\varepsilon) + \alpha \sum_{i=1}^{A} \log \sigma^i_\theta(s)] \to \max_\theta$

Soft Actor-Critic

- Actor $\pi_\theta(\,.\,|\,s)$, critics $Q_{\phi_1}(s,a)$, $Q_{\phi_2}(s,a)$, target critics $Q_{\phi_1^-}(s,a)$, $Q_{\phi_2^-}(s,a)$.

On each step:

- Observe s, choose $a \sim \pi_\theta(\,.\,|\,s)$, get $s', r, done$, put the transition into the buffer
- On the batch of transitions $(s_i, a, r_i, s_i', done_i)_{i=1}^{B}$ sampled from the replay buffer, perform:

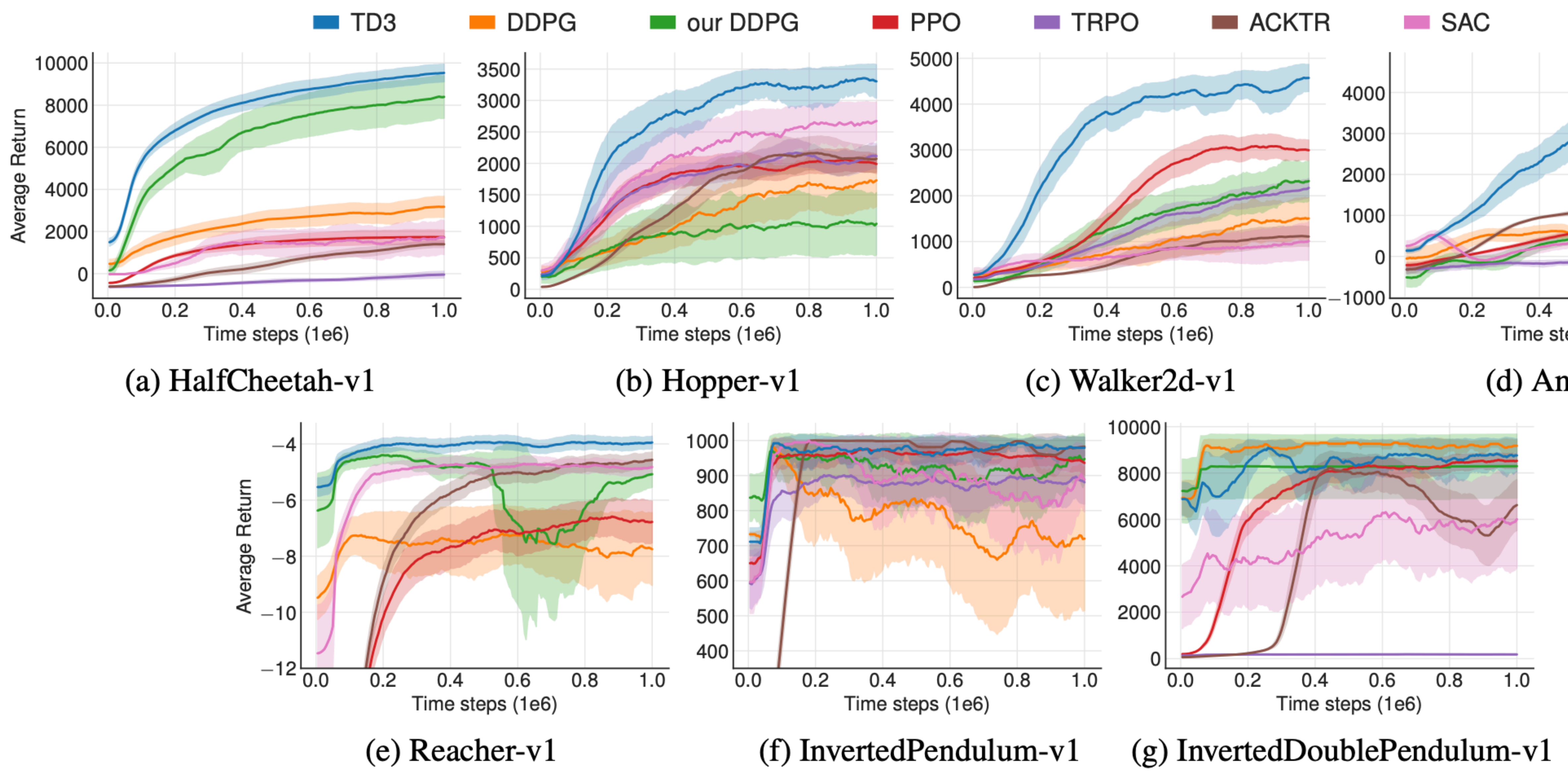
1. Soft Policy evaluation: $\frac{1}{B}\sum_{i=1}^{B}(y_i - Q_{\phi_k}(s_i, a_i))^2 \to \min\limits_{\phi_k}$, where
   $y_i = r_i + \gamma(1 - done_i)(\min\limits_{k=1,2} Q_{\phi_k^-}(s_i', a_i') - \alpha \log \pi(a_i'\,|\,s_i')),\ a_i' \sim \pi_\theta(\,.\,|\,s_i')$

2. Policy improvement: $\frac{1}{B}\sum_{i=1}^{B}\min\limits_{k=1,2} Q_{\phi_k}(s_i, a_\theta(s_i)) - \alpha \log \pi(a_\theta(s_i)) \to \max\limits_{\theta}$

   Where $a_\theta(s_i)$ is a sample from $\pi_\theta(\,.\,|\,s)$ which is differentiable wrt θ via reparametrisation trick

- Soft-update for the target critics

Comparison

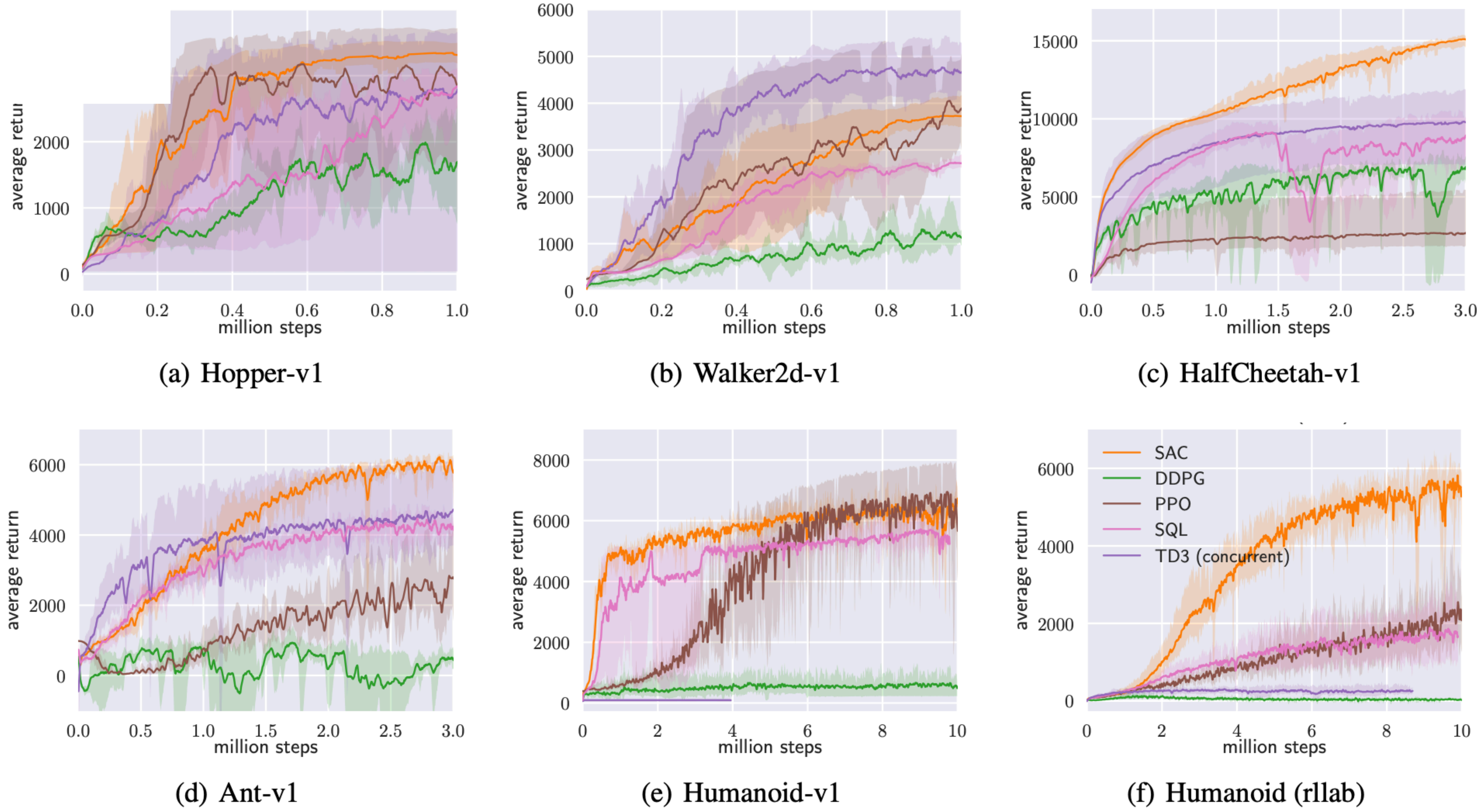


(a) HalfCheetah-v1 (b) Hopper-v1 (c) Walker2d-v1 (d) Ant-v1

(e) Reacher-v1 (f) InvertedPendulum-v1 (g) InvertedDoublePendulum-v1

TD3 paper

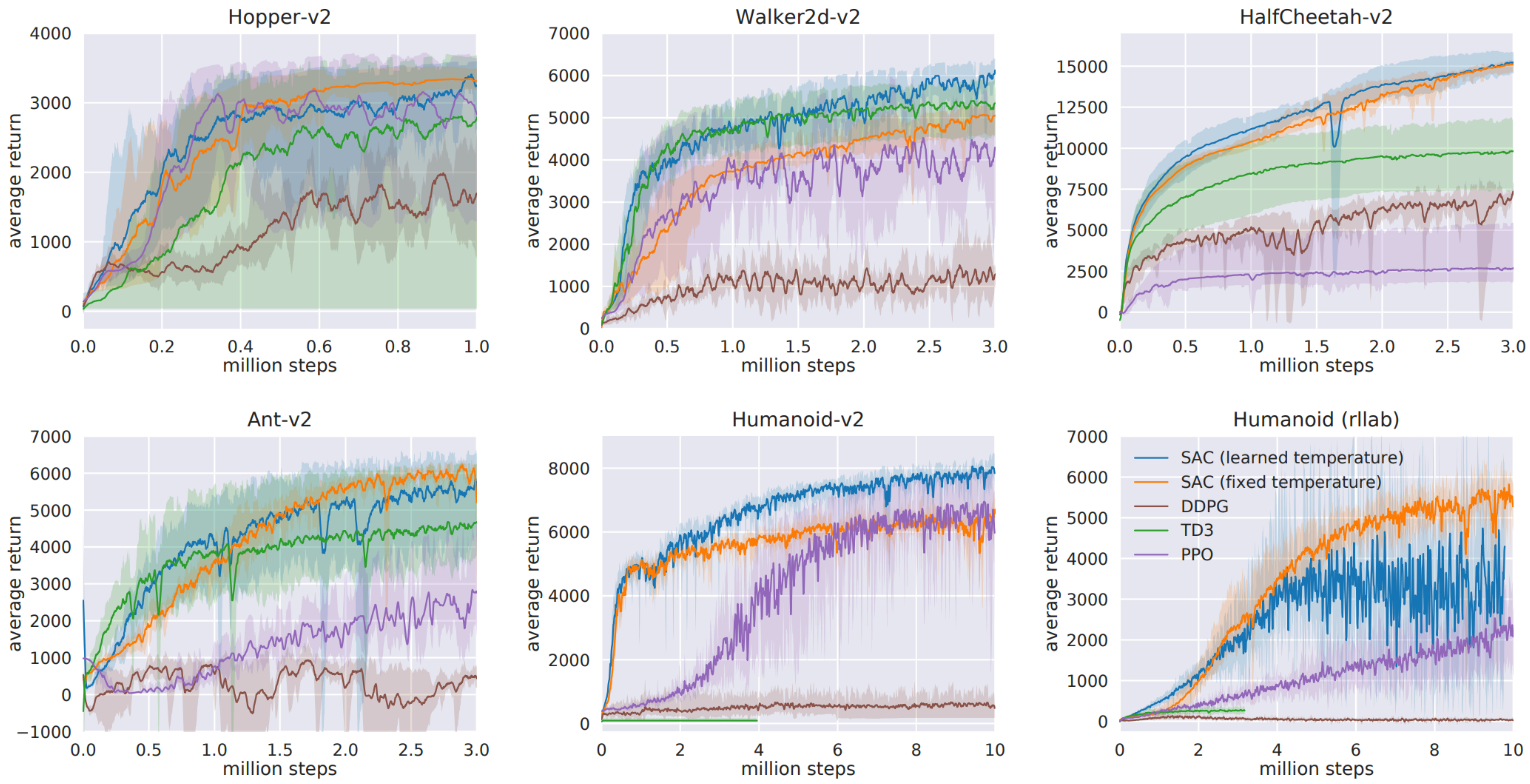
SAC paper



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

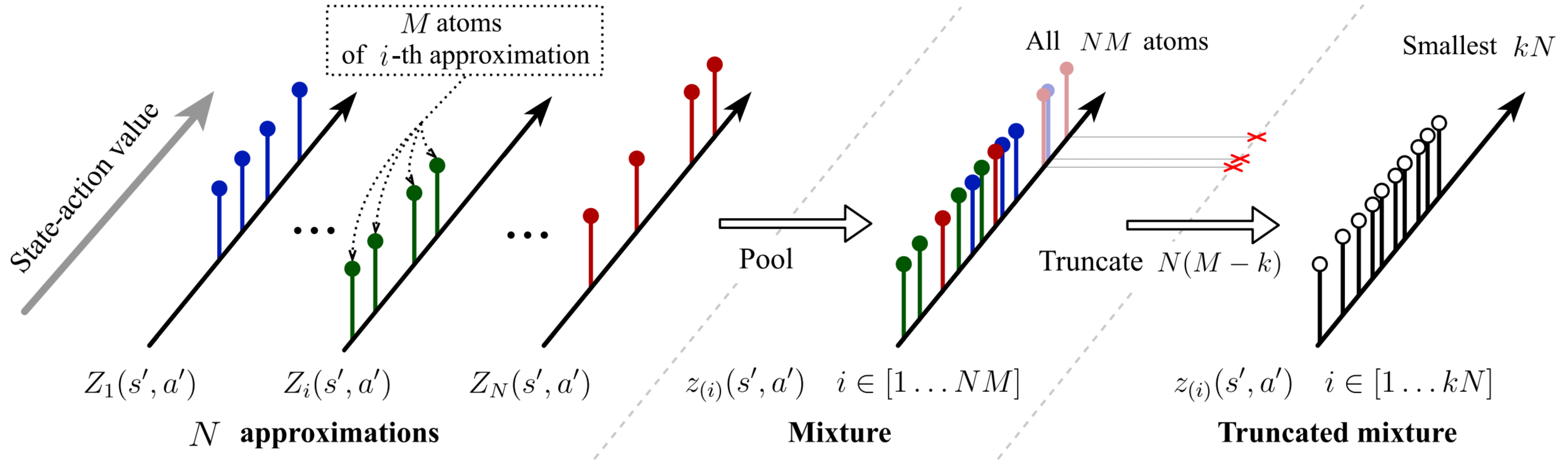
(d) Ant-v1 (e) Humanoid-v1 (f) Humanoid (rllab)

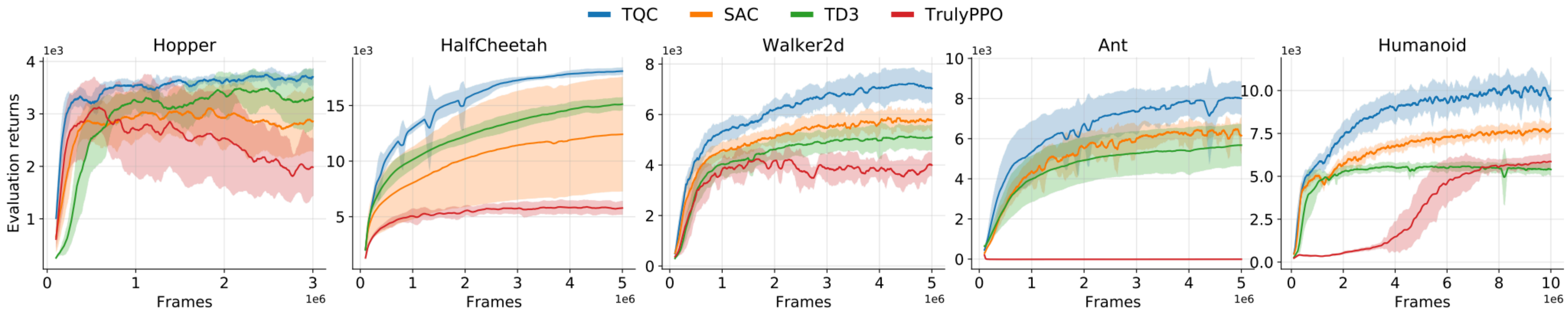
Comparison



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

Truncated Quantile Critics





Background

1. Reinforcement Learning Textbook (in Russian): 6
2. Soft Actor-Critic Algorithms and Applications
3. Lecture 19: Connection between Inference and Control

Thank you for your attention!