RL: Policy Gradient -- Actor-Critic Algos

How do we decide what to do?

- Emotions/Intuition

$$V_t(s) \qquad Q_t(s, a)$$

- Thinking

$$S_{t+1} = M(S_t, A_t, \theta)$$

- Reflexes/Habits

$$A_t = \pi(S_t, \theta)$$

Policy Approximation

$$\pi(a|s,\boldsymbol{\theta})$$

We want to learn this directly!

- Policy = a function from state to action
  - How does the agent select actions?
  - In such a way that it can be affected by learning?
  - In such a way as to assure exploration?
- Approximation: there are too many states and/or actions to represent all policies
  - To handle large/continuous action spaces

Episodic policy gradients algorithm

Policy Gradient Theorem (PGT):

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\pi) = \mathbb{E}_\pi[\sum_{t=0}^{T} \gamma^t q_\pi(S_t, A_t) \nabla_{\boldsymbol{\theta}} \log \pi(A_t|S_t)]$$

- We can sample this, given a whole episode
- Typically, people pull out the sum, and split up this into separate gradients, e.g.,

$$\Delta\boldsymbol{\theta}_t = \gamma^t G_t \nabla_{\boldsymbol{\theta}} \log \pi(A_t|S_t)$$

  such that $\mathbb{E}_\pi[\sum_t \Delta\boldsymbol{\theta}_t] = \nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\pi)$
- Typically, people ignore the γ^t term, use $\Delta\boldsymbol{\theta}_t = G_t \nabla_{\boldsymbol{\theta}} \log \pi(A_t|S_t)$
- This is actually okay-ish — we just partially pretend on each step that we could have started an episode in that state instead. Or if we use γ=1, this is also ok.
  (alternatively, view it as a slightly biased gradient)

REINFORCE (Monte-Carlo)

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\pi) = \mathbb{E}_{\pi}[\sum_{t=0}^{T} (\gamma^t G_t) \nabla_{\boldsymbol{\theta}} \log \pi(A_t | S_t)]$$

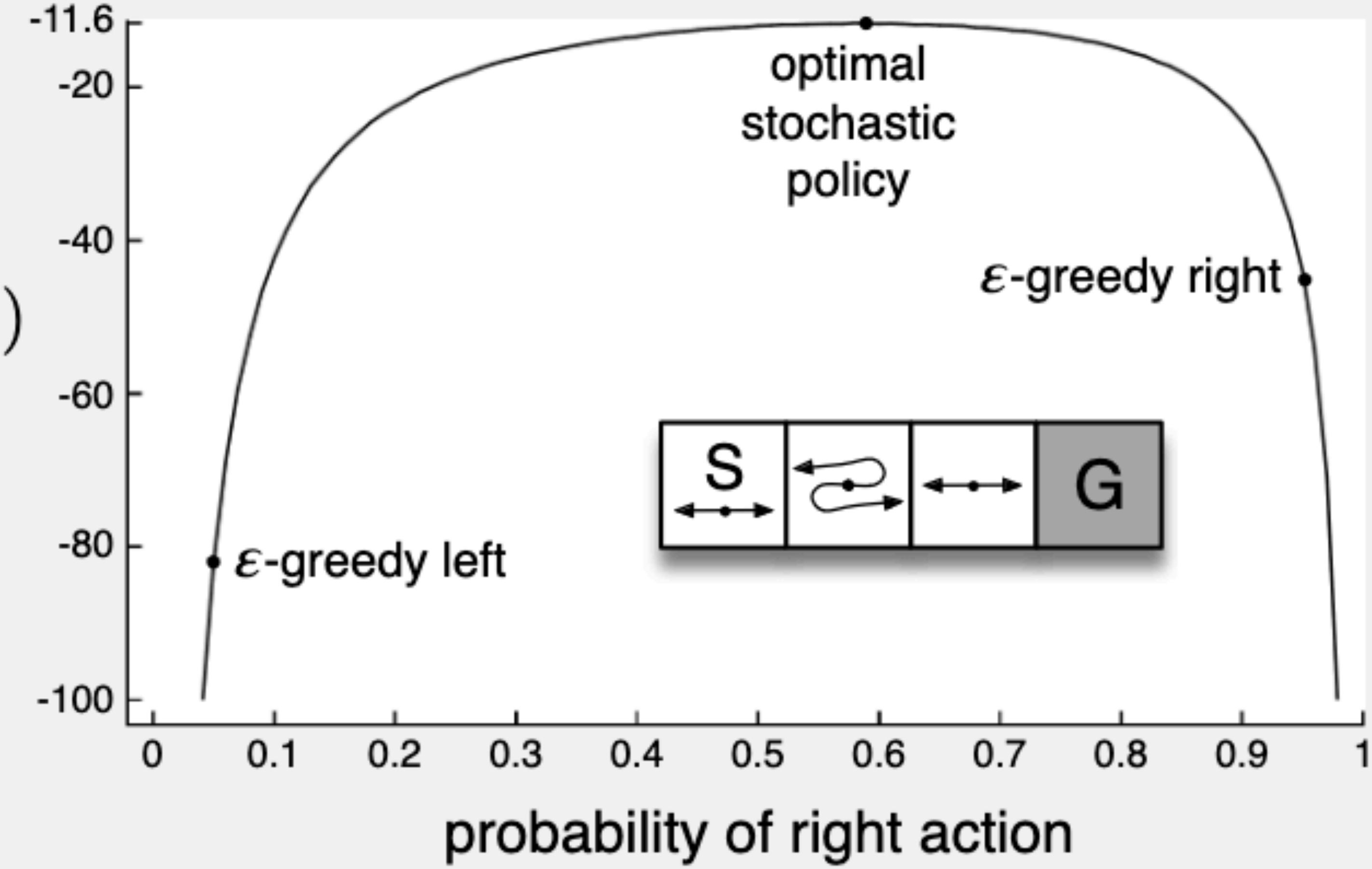
REINFORCE: Monte-Carlo Policy-Gradient Control (episodic) for π_*

Input: a differentiable policy parameterization $\pi(a|s, \boldsymbol{\theta})$
Algorithm parameter: step size $\alpha > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ (e.g., to $\mathbf{0}$)

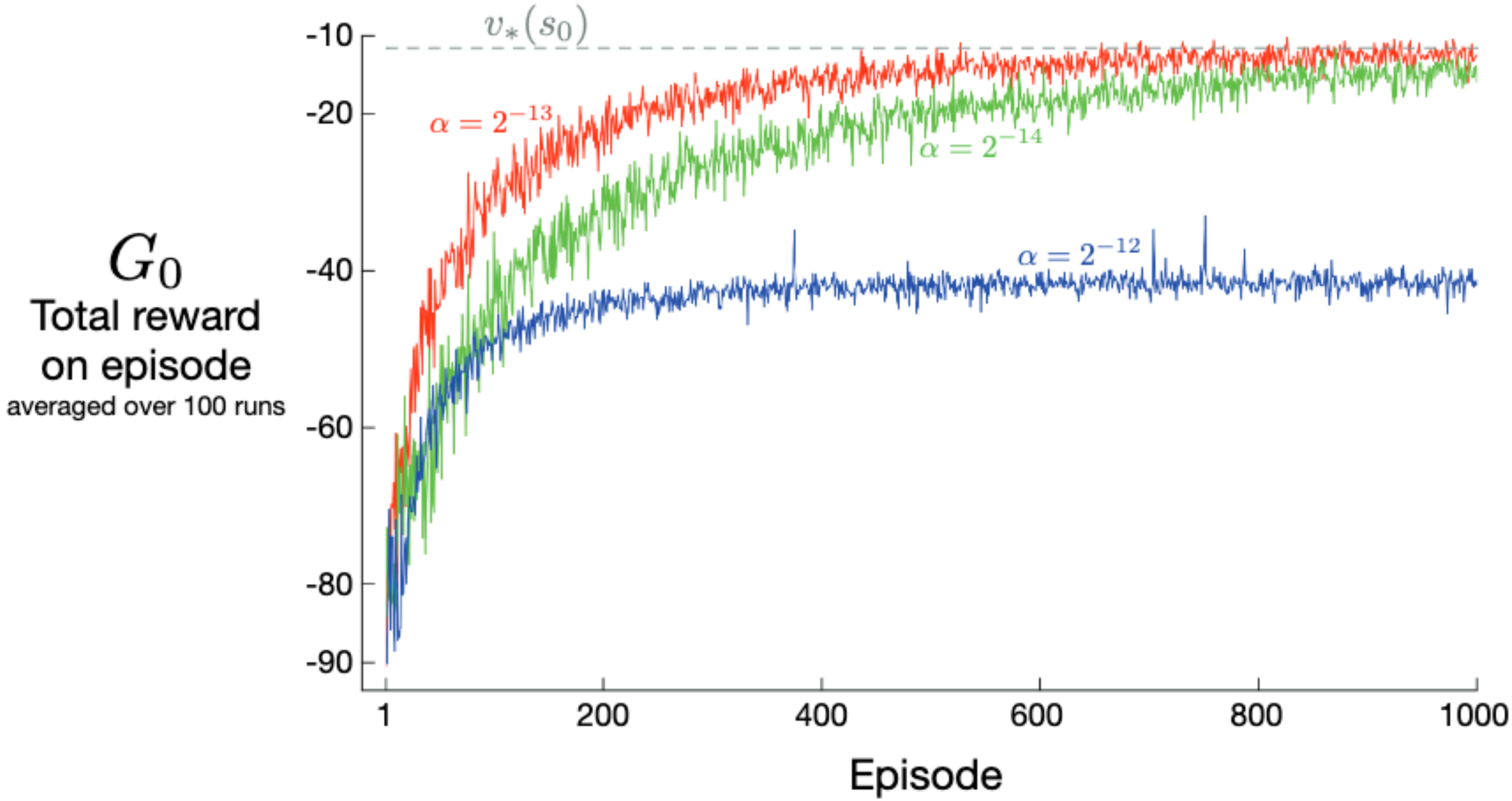
Loop forever (for each episode):
  Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot, \boldsymbol{\theta})$
  Loop for each step of the episode $t = 0, 1, \ldots, T-1$:
    $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ (G_t)
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \gamma^t G \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta})$

Example: REINFORCE

$$J(\boldsymbol{\theta}) = v_{\pi_{\boldsymbol{\theta}}}(\mathsf{S})$$



Example: REINFORCE



Improvements to REINFORCE

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\pi) = \mathbb{E}_{\pi}[\sum_{t=0}^{T} \left(\gamma^t G_t\right) \nabla_{\boldsymbol{\theta}} \log \pi(A_t | S_t)]$$

- Can we use our "trick" $\mathbb{E}\left(b(s)\nabla_\theta \log(\pi(a\,|\,s,\theta))\right) = 0$ to improve REINFORCE?

Reducing Variance:

$$\bar{X} = \mathbb{E}(X) = 0$$

X, Y are two random variables.

$$\bar{Y} = \mathbb{E}(Y) \neq 0$$

Using samples of X, Y, I want to estimate: $\mathbb{E}(YX) \equiv J$

Reducing Variance:

$$\bar{X} = \mathbb{E}(X) = 0$$

X, Y are two random variables.

$$\bar{Y} = \mathbb{E}(Y) \neq 0$$

Using samples of X, Y, I want to estimate: $\mathbb{E}(YX) \equiv J$

$$J \approx \frac{1}{N} \sum_{i}^{N} Y_i X_i$$

Reducing Variance:

$$\bar{X} = \mathbb{E}(X) = 0$$

X, Y are two random variables.

$$\bar{Y} = \mathbb{E}(Y) \neq 0$$

Using samples of X, Y, I want to estimate: $\mathbb{E}(YX) \equiv J$

$$J \approx \frac{1}{N} \sum_{i}^{N} Y_i X_i$$

Can I do it with less variance??

Reducing Variance:

$$\bar{X} = \mathbb{E}(X) = 0$$

X, Y are two random variables.

$$\bar{Y} = \mathbb{E}(Y) \neq 0$$

Using samples of X, Y, I want to estimate: $\mathbb{E}(YX) \equiv J$

$$\mathbb{E}(YX) = \mathbb{E}\left[(Y - \bar{Y})X + \bar{Y}X\right] = \mathbb{E}\left[(Y - \bar{Y})X\right] + \bar{Y}\mathbb{E}\left[X\right]$$

Reducing Variance:

$$\bar{X} = \mathbb{E}(X) = 0$$

X, Y are two random variables.

$$\bar{Y} = \mathbb{E}(Y) \neq 0$$

Using samples of X, Y, I want to estimate: $\mathbb{E}(YX) \equiv J$

$$\mathbb{E}(YX) = \mathbb{E}\left[(Y - \bar{Y})X + \bar{Y}X\right] = \mathbb{E}\left[(Y - \bar{Y})X\right] + \bar{Y}\overbrace{\mathbb{E}\left[X\right]}^{0}$$

Reducing Variance:

X, Y are two random variables.

$$\bar{X} = \mathbb{E}(X) = 0$$

$$\bar{Y} = \mathbb{E}(Y) \neq 0$$

Using samples of X, Y, I want to estimate: $\mathbb{E}(YX) \equiv J$

$$\mathbb{E}(YX) = \mathbb{E}\left[(Y - \bar{Y})X + \bar{Y}X\right] = \mathbb{E}\left[(Y - \bar{Y})X\right] + \bar{Y}\underbrace{\mathbb{E}\left[X\right]}_{0}$$

$$J \approx \frac{1}{N}\sum_{i}^{N}(Y_i - \bar{Y})X_i + \frac{\bar{Y}}{N}\sum_{i}^{N}X_i$$

Will eventually go to 0, just adds noise!

Improvements to REINFORCE

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\pi) = \mathbb{E}_{\pi}[\sum_{t=0}^{T} \left(\gamma^{t} G_{t}\right) \nabla_{\boldsymbol{\theta}} \log \pi(A_{t} | S_{t})]$$

- Can we use our "trick" $\mathbb{E}\left(b(s)\nabla_\theta \log(\pi(a\,|\,s,\theta))\right) = 0$ to improve REINFORCE?

$$\nabla_\theta J_\theta(\pi) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \left(G_t - \bar{G}\right) \nabla_\theta \log(\pi)\right]$$

Improvements to REINFORCE

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\pi) = \mathbb{E}_{\pi}[\sum_{t=0}^{T} \left(\gamma^{t} G_{t}\right) \nabla_{\boldsymbol{\theta}} \log \pi(A_{t}|S_{t})]$$

- Can we use our "trick" $\mathbb{E}\left(b(s)\nabla_\theta \log(\pi(a\,|\,s,\theta))\right) = 0$ to improve REINFORCE?

$$\nabla_\theta J_\theta(\pi) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \left(G_t - \bar{G}\right) \nabla_\theta \log(\pi)\right]$$

$$\nabla_\theta J_\theta(\pi) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \left(q_\pi(S_t, A_t) - v_\pi(S_t)\right) \nabla_\theta \log(\pi)\right]$$

Improvements to REINFORCE

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\pi) = \mathbb{E}_{\pi}[\sum_{t=0}^{T} \left(\gamma^t G_t\right) \nabla_{\boldsymbol{\theta}} \log \pi(A_t|S_t)]$$

- Can we use our "trick" $\mathbb{E}\left(b(s)\nabla_\theta \log(\pi(a\,|\,s,\theta))\right) = 0$ to improve REINFORCE?

$$\nabla_\theta J_\theta(\pi) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \left(G_t - \bar{G}\right) \nabla_\theta \log(\pi)\right]$$

$$\nabla_\theta J_\theta(\pi) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \left(q_\pi(S_t, A_t) - v_\pi(S_t)\right) \nabla_\theta \log(\pi)\right]$$

Advantage

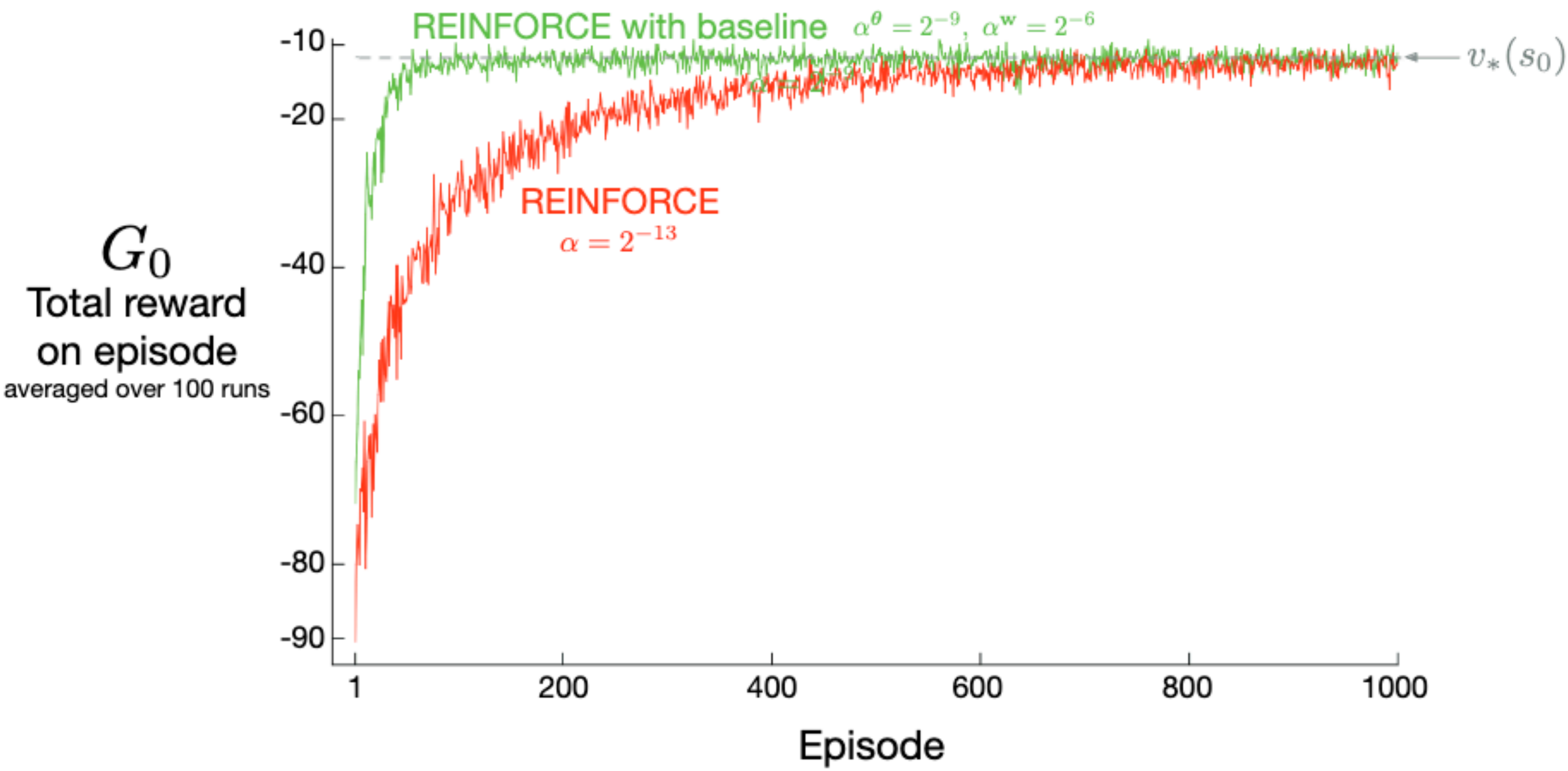
REINFORCE with baseline:

REINFORCE with Baseline (episodic), for estimating $\pi_{\boldsymbol{\theta}} \approx \pi_*$

Input: a differentiable policy parameterization $\pi(a|s, \boldsymbol{\theta})$
Input: a differentiable state-value function parameterization $\hat{v}(s,\mathbf{w})$
Algorithm parameters: step sizes $\alpha^{\boldsymbol{\theta}} > 0$, $\alpha^{\mathbf{w}} > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ and state-value weights $\mathbf{w} \in \mathbb{R}^{d}$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):
  Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot, \boldsymbol{\theta})$
  Loop for each step of the episode $t = 0, 1, \ldots, T-1$:
    $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ (G_t)
    $\delta \leftarrow G - \hat{v}(S_t,\mathbf{w})$
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}} \delta \nabla \hat{v}(S_t,\mathbf{w})$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}} \gamma^t \delta \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta})$

REINFORCE with baseline:



Actor-Critic Algorithms

- ACTOR: policy π
- CRITIC: value fct V (or Q)

Actor-Critic Algorithms

- ACTOR: policy π
- CRITIC: value fct V (or Q)

Policy Gradient Theorem:

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\pi) = \mathbb{E}_{\pi}[\sum_{t=0}^{T} (\gamma^t G_t) \nabla_{\boldsymbol{\theta}} \log \pi(A_t | S_t)]$$

Actor-Critic Algorithms

- ACTOR: policy π
- CRITIC: value fct V (or Q)

Policy Gradient Theorem:

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\pi) = \mathbb{E}_{\pi}[\sum_{t=0}^{T} (\gamma^t G_t) \nabla_{\boldsymbol{\theta}} \log \pi(A_t | S_t)]$$

REINFORCE Estimates G with Monte-Carlo

Actor-Critic Algorithms

- ACTOR: policy π
- CRITIC: value fct V (or Q)

Policy Gradient Theorem:

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\pi) = \mathbb{E}_{\pi}[\sum_{t=0}^{T} (\gamma^t G_t) \nabla_{\boldsymbol{\theta}} \log \pi(A_t|S_t)]$$

Actor-Critic: use V and/or Q to estimate G, e.g. TD(0)

Actor-Critic 1-step TD / TD(0) estimate:

Policy Gradient Theorem:

$$\nabla_\theta J_\theta(\pi) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \underbrace{\left(q_\pi(S_t, A_t) - v_\pi(S_t)\right)}_{\text{Advantage}} \nabla_\theta \log(\pi)\right]$$

$$\begin{aligned}
\boldsymbol{\theta}_{t+1} &\doteq \boldsymbol{\theta}_t + \alpha\Big(G_{t:t+1} - \hat{v}(S_t,\mathbf{w})\Big)\frac{\nabla\pi(A_t|S_t,\boldsymbol{\theta}_t)}{\pi(A_t|S_t,\boldsymbol{\theta}_t)} \\
&= \boldsymbol{\theta}_t + \alpha\Big(R_{t+1} + \gamma\hat{v}(S_{t+1},\mathbf{w}) - \hat{v}(S_t,\mathbf{w})\Big)\frac{\nabla\pi(A_t|S_t,\boldsymbol{\theta}_t)}{\pi(A_t|S_t,\boldsymbol{\theta}_t)} \\
&= \boldsymbol{\theta}_t + \alpha\delta_t\frac{\nabla\pi(A_t|S_t,\boldsymbol{\theta}_t)}{\pi(A_t|S_t,\boldsymbol{\theta}_t)}.
\end{aligned}$$

Actor-Critic 1-step TD / TD(0) estimate:

Policy Gradient Theorem:

$$\nabla_\theta J_\theta(\pi) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \left(q_\pi(S_t, A_t) - v_\pi(S_t)\right) \nabla_\theta \log(\pi)\right]$$

Advantage

One-step Actor–Critic (episodic), for estimating $\pi_{\boldsymbol{\theta}} \approx \pi_*$

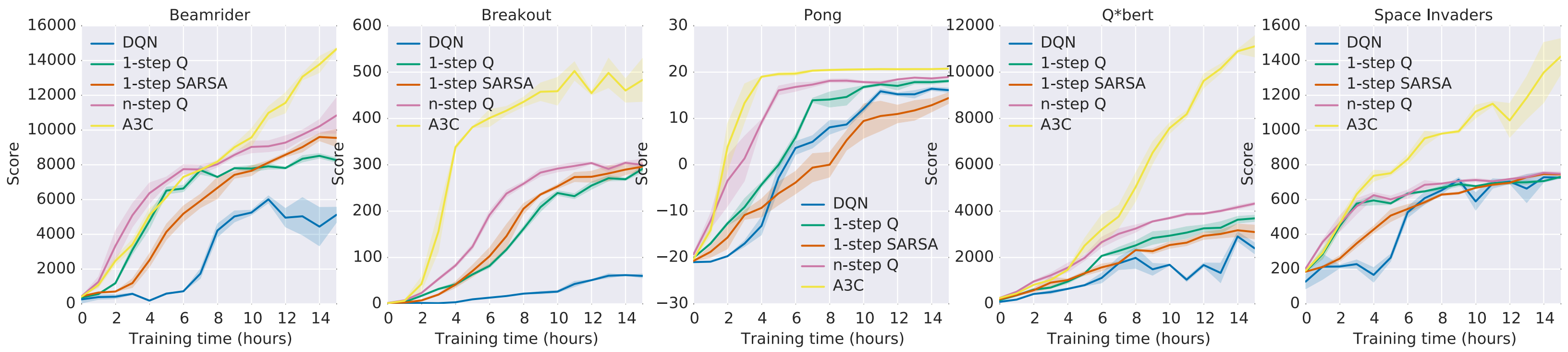
Input: a differentiable policy parameterization $\pi(a|s, \boldsymbol{\theta})$
Input: a differentiable state-value function parameterization $\hat{v}(s,\mathbf{w})$
Parameters: step sizes $\alpha^{\boldsymbol{\theta}} > 0$, $\alpha^{\mathbf{w}} > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ and state-value weights $\mathbf{w} \in \mathbb{R}^d$ (e.g., to $\mathbf{0}$)
Loop forever (for each episode):
  Initialize S (first state of episode)
  $I \leftarrow 1$
  Loop while S is not terminal (for each time step):
    $A \sim \pi(\cdot|S, \boldsymbol{\theta})$
    Take action A, observe S', R
    $\delta \leftarrow R + \gamma\hat{v}(S',\mathbf{w}) - \hat{v}(S,\mathbf{w})$ (if S' is terminal, then $\hat{v}(S',\mathbf{w}) \doteq 0$)
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}} \delta \nabla \hat{v}(S,\mathbf{w})$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}} I \delta \nabla \ln \pi(A|S, \boldsymbol{\theta})$
    $I \leftarrow \gamma I$
    $S \leftarrow S'$

A3C: Asynchronous Advantage Actor Critic:

$$\nabla_\theta J_\theta(\pi) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \left(\underbrace{q_\pi(S_t, A_t) - v_\pi(S_t)}_{\text{Advantage}}\right) \nabla_\theta \log(\pi)\right]$$

A3C: Asynchronous Advantage Actor Critic:

$$\nabla_\theta J_\theta(\pi) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \left(\underbrace{q_\pi(S_t, A_t) - v_\pi(S_t)}_{\textbf{Advantage}}\right) \nabla_\theta \log(\pi)\right]$$



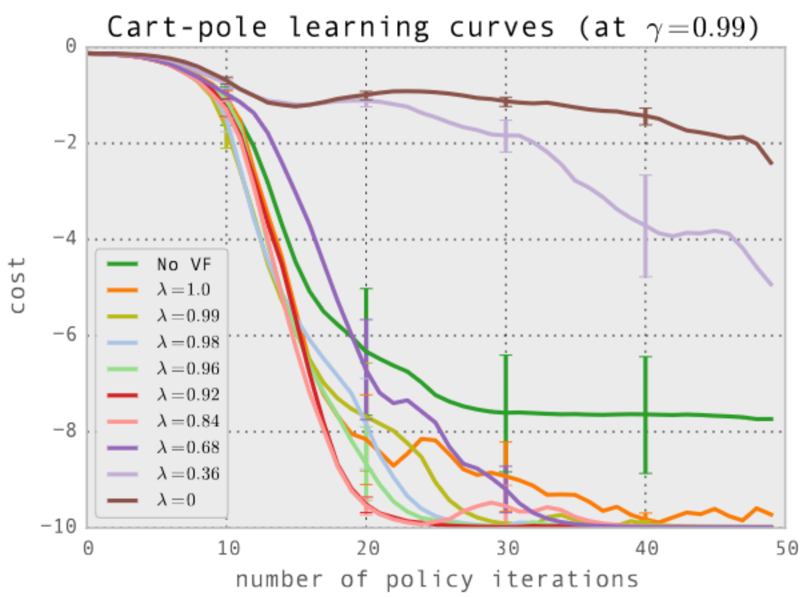
GAE: Generalized Advantage Estimation

- Use Advantage (i.e. G - V(S))
- Use TD(λ) target for G

https://arxiv.org/abs/1506.02438

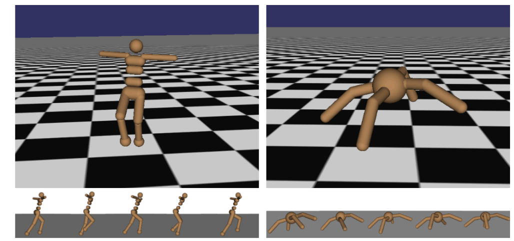
GAE: Generalized Advantage Estimation

- Use Advantage (i.e. G - V(S))
- Use TD(λ) target for G

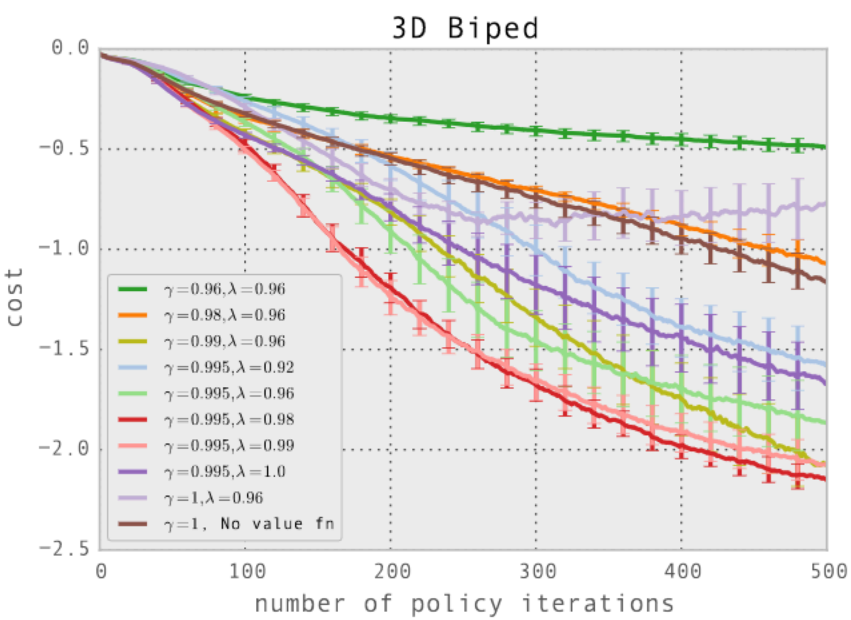


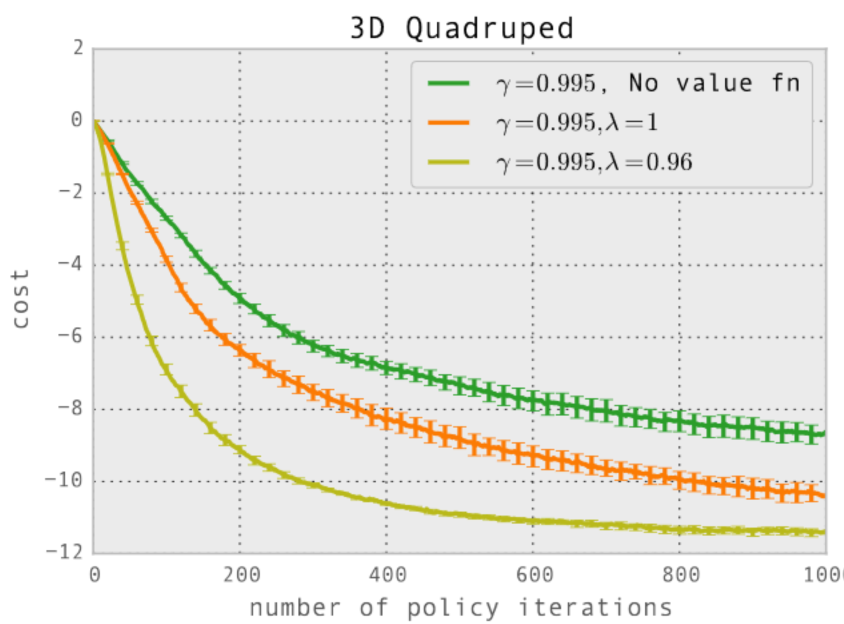
https://arxiv.org/abs/1506.02438

GAE: Generalized Advantage Estimation



- Use Advantage (i.e. G - V(S))
- Use TD(λ) target for G





https://arxiv.org/abs/1506.02438

What about if we want a Deterministic Policy?

We can't use the Policy Gradient Theorem :

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\pi) = \mathbb{E}_{\pi}[\sum_{t=0}^{T} (\gamma^t G_t) \nabla_{\boldsymbol{\theta}} \log \pi(A_t | S_t)]$$

How can we estimate $\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\pi)$? When $A = \pi(S, \theta)$

What about if we want a Deterministic Policy?

We can't use the Policy Gradient Theorem :

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\pi) = \mathbb{E}_{\pi}[\sum_{t=0}^{T} (\gamma^{t} G_{t}) \nabla_{\boldsymbol{\theta}} \log \pi(A_t | S_t)]$$

How can we estimate $\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\pi)$? When $A = \pi(S, \theta)$

$$J_{\theta}(\pi \,|\, S_0 = S) = q_{\pi}(\pi(S), S) \approx Q_{\pi}(\pi(S, \theta), S)$$

What about if we want a Deterministic Policy?

How can we estimate $\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\pi)$? When $A = \pi(S, \theta)$

$$A = (a_1, \ldots, a_m),\ \pi = (\pi_1, \ldots, \pi_m)$$

$$J_\theta(\pi \,|\, S_0 = S) = q_\pi(\pi(S), S) \approx Q_\pi(\pi(S, \theta), S)$$

What about if we want a Deterministic Policy?

How can we estimate $\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\pi)$? When $A = \pi(S, \theta)$

$A = (a_1, \ldots, a_m),\ \pi = (\pi_1, \ldots, \pi_m)$

$$J_\theta(\pi \mid S_0 = S) = q_\pi(\pi(S), S) \approx Q_\pi(\pi(S, \theta), S)$$

$$\nabla_\theta J_\theta(\pi \mid S_0 = S) \approx \nabla_\theta Q_\pi(\pi(S, \theta), S)$$

What about if we want a Deterministic Policy?

How can we estimate $\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\pi)$? When $A = \pi(S, \theta)$

$A = (a_1, \ldots, a_m)$, $\pi = (\pi_1, \ldots, \pi_m)$

$$J_\theta(\pi \mid S_0 = S) = q_\pi(\pi(S), S) \approx Q_\pi(\pi(S, \theta), S)$$

$$\nabla_\theta J_\theta(\pi \mid S_0 = S) \approx \nabla_\theta Q_\pi(\pi(S, \theta), S)$$

$$= \sum_i^m \frac{\partial Q_\pi(A = \pi(S, \theta), S)}{\partial a_i} \nabla_\theta \pi_i(S, \theta)$$

What about if we want a Deterministic Policy?

How can we estimate $\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\pi)$? When $A = \pi(S, \theta)$

$A = (a_1, \ldots, a_m),\ \pi = (\pi_1, \ldots, \pi_m)$

$$J_\theta(\pi \mid S_0 = S) = q_\pi(\pi(S), S) \approx Q_\pi(\pi(S, \theta), S)$$

$$\begin{aligned}
\nabla_\theta J_\theta(\pi \mid S_0 = S) &\approx \nabla_\theta Q_\pi(\pi(S, \theta), S) \\
&= \sum_i^m \frac{\partial Q_\pi(A = \pi(S, \theta), S)}{\partial a_i} \nabla_\theta \pi_i(S, \theta) \\
&= \nabla_A Q_\pi \left(A = \pi(S, \theta), S\right) \nabla_\theta \pi(S, \theta)
\end{aligned}$$

Deterministic Policy Gradient:

How can we estimate $\boldsymbol{\nabla_\theta J_\theta(\pi)}$? When $A = \pi(S, \theta)$

$A = (a_1, \ldots, a_m),\ \pi = (\pi_1, \ldots, \pi_m)$

$$\nabla_\theta J_\theta(\pi \,|\, S_0 = S) \approx \sum_i^m \frac{\partial Q_\pi(A = \pi(S,\theta), S)}{\partial a_i} \nabla_\theta \pi_i(S,\theta)$$

$$= \nabla_A Q_\pi \left(A = \pi(S,\theta), S\right) \nabla_\theta \pi(S,\theta)$$

http://proceedings.mlr.press/v32/silver14.pdf

Deterministic Policy Gradient (on Continuous Control Tasks):

Deterministic Policy Gradient Algorithms

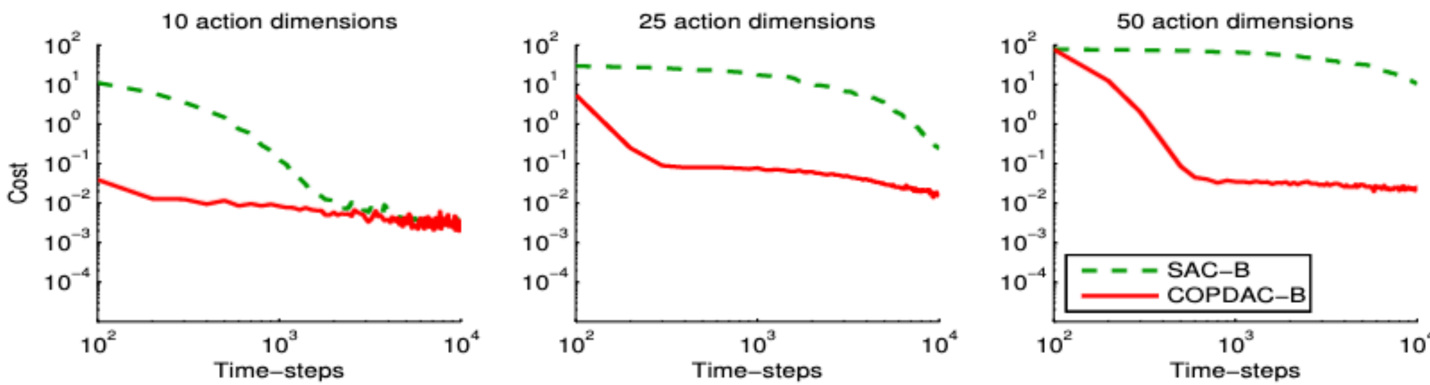
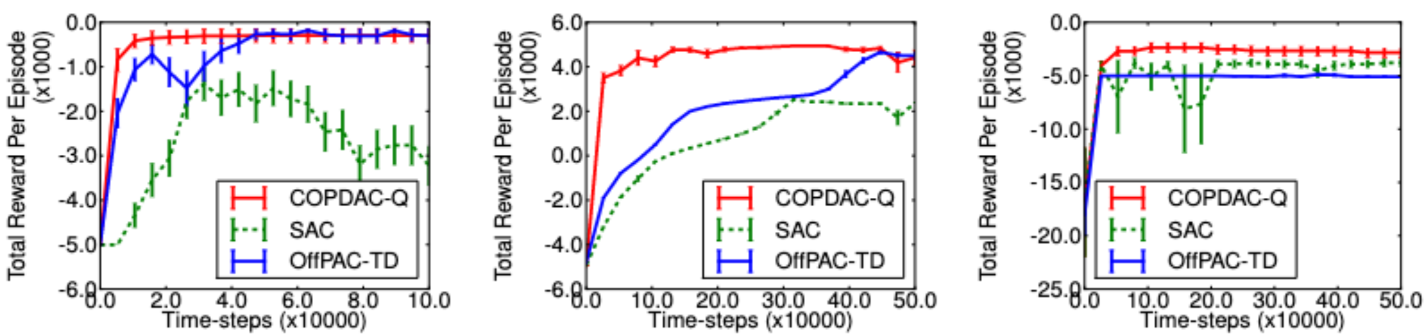


Figure 1. Comparison of stochastic actor-critic (SAC-B) and deterministic actor-critic (COPDAC-B) on the continuous bandit task.



(a) Mountain Car (b) Pendulum (c) 2D Puddle World

Figure 2. Comparison of stochastic on-policy actor-critic (SAC), stochastic off-policy actor-critic (OffPAC), and deterministic off-policy actor-critic (COPDAC) on continuous-action reinforcement learning. Each point is the average test performance of the mean policy.

http://proceedings.mlr.press/v32/silver14.pdf

Deterministic Policy Gradient (on Continuous Control Tasks):

Actor Critic with stochastic policy

Deterministic Policy Gradient

Deterministic Policy Gradient Algorithms

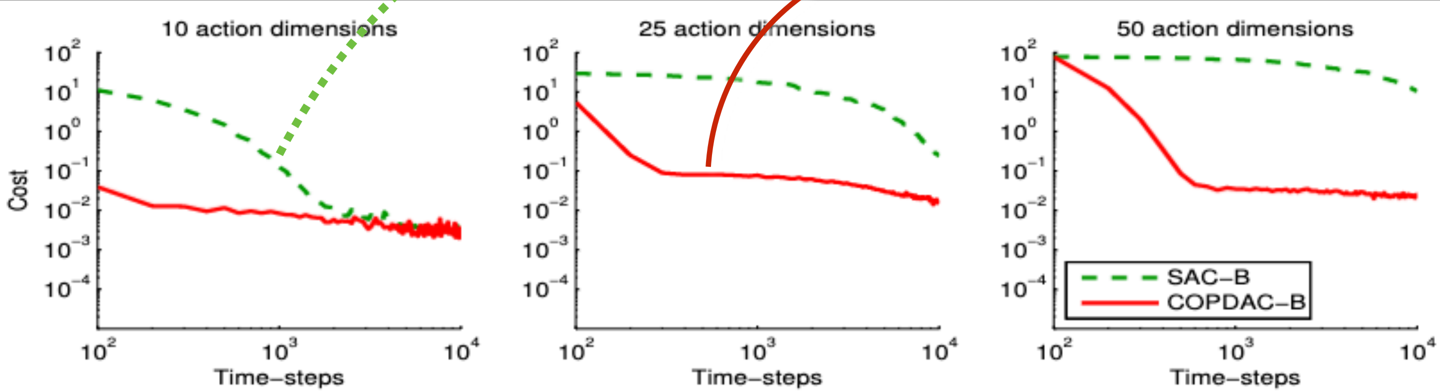
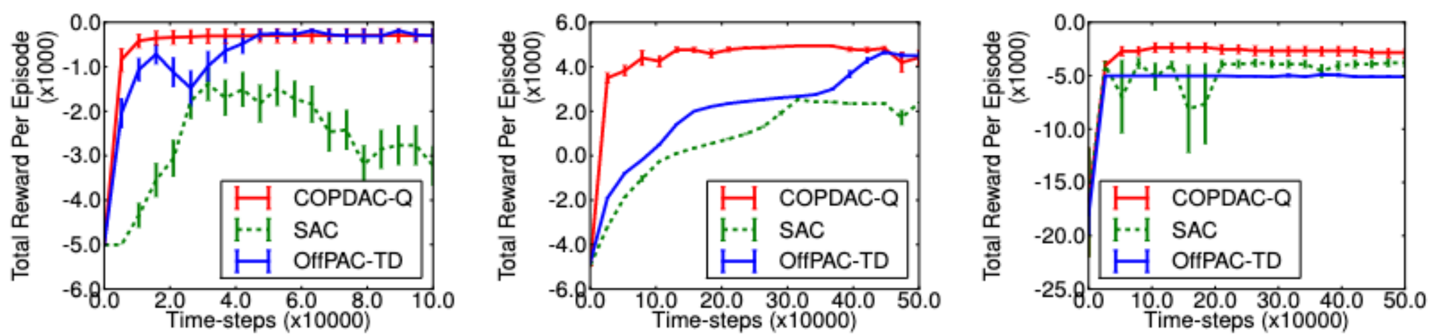


Figure 1. Comparison of stochastic actor-critic (SAC-B) and deterministic actor-critic (COPDAC-B) on the continuous bandit task.



(a) Mountain Car (b) Pendulum (c) 2D Puddle World

Figure 2. Comparison of stochastic on-policy actor-critic (SAC), stochastic off-policy actor-critic (OffPAC), and deterministic off-policy actor-critic (COPDAC) on continuous-action reinforcement learning. Each point is the average test performance of the mean policy.

http://proceedings.mlr.press/v32/silver14.pdf

Deep Deterministic Policy Gradient (DDPG):

Algorithm 1 DDPG algorithm

Randomly initialize critic network $Q(s, a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ.
Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer R
for episode = 1, M **do**
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial observation state s_1
  for t = 1, T **do**
    Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise
    Execute action a_t and observe reward r_t and observe new state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in R
    Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from R
    Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
    Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$
    Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s_i}$$

    Update the target networks:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$

$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$

  end for
end for

https://arxiv.org/pdf/1509.02971.pdf

Conclusion

- Policy Gradient Theorem: $\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}(\pi) = \mathbb{E}_{\pi}[\sum_{t=0}^{T} (\gamma^t G_t) \nabla_{\boldsymbol{\theta}} \log \pi(A_t | S_t)]$
- REINFORCE: PGT + MC for estimate of G
- Actor-Critic: PGT + V,Q for estimate of G
- Deterministic Policy Gradient: $\nabla_\theta J_\theta(\pi | S_0 = S) \approx \nabla_\theta Q_\pi(\pi(S, \theta), S)$