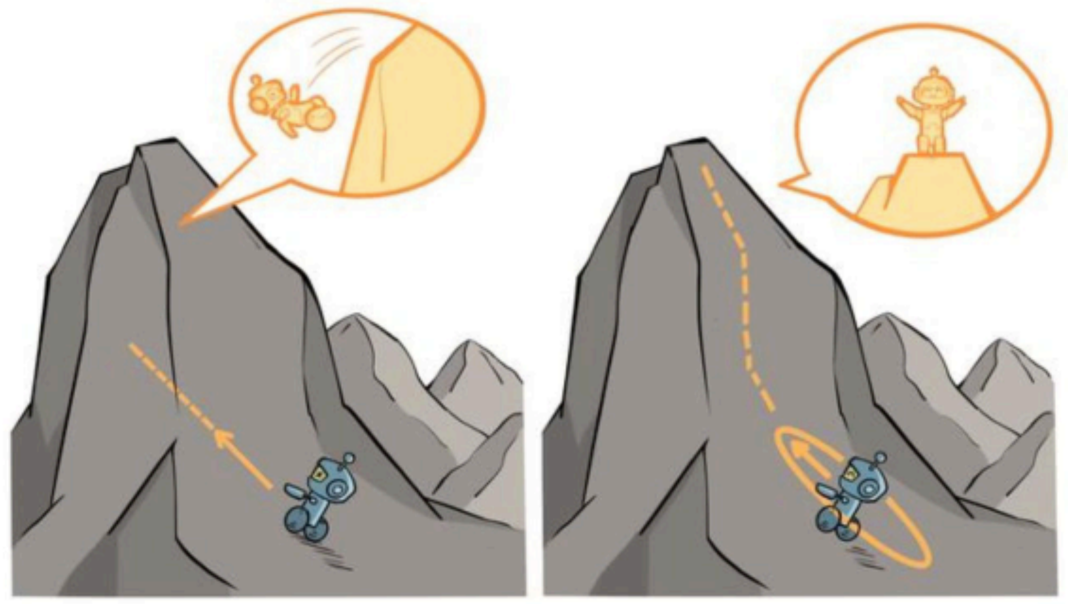
Trust Region Policy Optimization



Nicolas Espinosa Dice

Slides adapted from Wen Sun
(with inspiration from Benjamin Eysenbach)

Improving Policy Gradient

Lecture 10: **Policy gradient**

Lecture 11: **Variance Reduction** via advantage estimation

Lecture 12 (today!): **Leverage the geometry** via Natural Policy Gradient (NPG)

Recap Policy Gradient

$$J(\pi_\theta) = \mathbb{E}\left[\sum_{h=0}^{\infty} \gamma^h r(s_h, a_h) \,|\, s_0 \sim \mu, a \sim \pi_\theta\right]$$

The most commonly used formulation:

$$\nabla_\theta J(\pi_{\theta_t}) = \mathbb{E}_{s,a \sim d_\mu^{\pi_{\theta_t}}}\left[\nabla_\theta \ln \pi_{\theta_t}(a \,|\, s) A^{\pi_{\theta_t}}(s, a)\right]$$

Algorithm: Stochastic Gradient Ascent

Policy Parameterization

Recall that we consider parameterized policy $\pi_\theta(\cdot \mid s) \in \Delta(A), \forall s$

1. Softmax linear Policy

Feature vector $\phi(s, a) \in \mathbb{R}^d$, and parameter $\theta \in \mathbb{R}^d$

$$\pi_\theta(a \mid s) = \frac{\exp(\theta^\top \phi(s, a))}{\sum_{a'} \exp(\theta^\top \phi(s, a'))}$$

2. Neural Policy:

Neural network $f_\theta : S \times A \mapsto \mathbb{R}$

$$\pi_\theta(a \mid s) = \frac{\exp(f_\theta(s, a))}{\sum_{a'} \exp(f_\theta(s, a'))}$$

Outline

1. Motivation behind trust-region policy optimization

2. Quick intro on KL-divergence

3. A Trust-Region Formulation for Policy Optimization

4. Algorithm: Natural Policy Gradient

Two Observations

Observation 1: Policy gradient estimates have high variance
Observation 2: Small changes in policy's parameters can lead to *large changes in policy*

Today's Question

Can we optimize the policy's parameters while considering the policy's change?

Intuition Behind Observation #2

Observation 2: Small changes in policy's parameters can lead to *large changes in policy*

Example

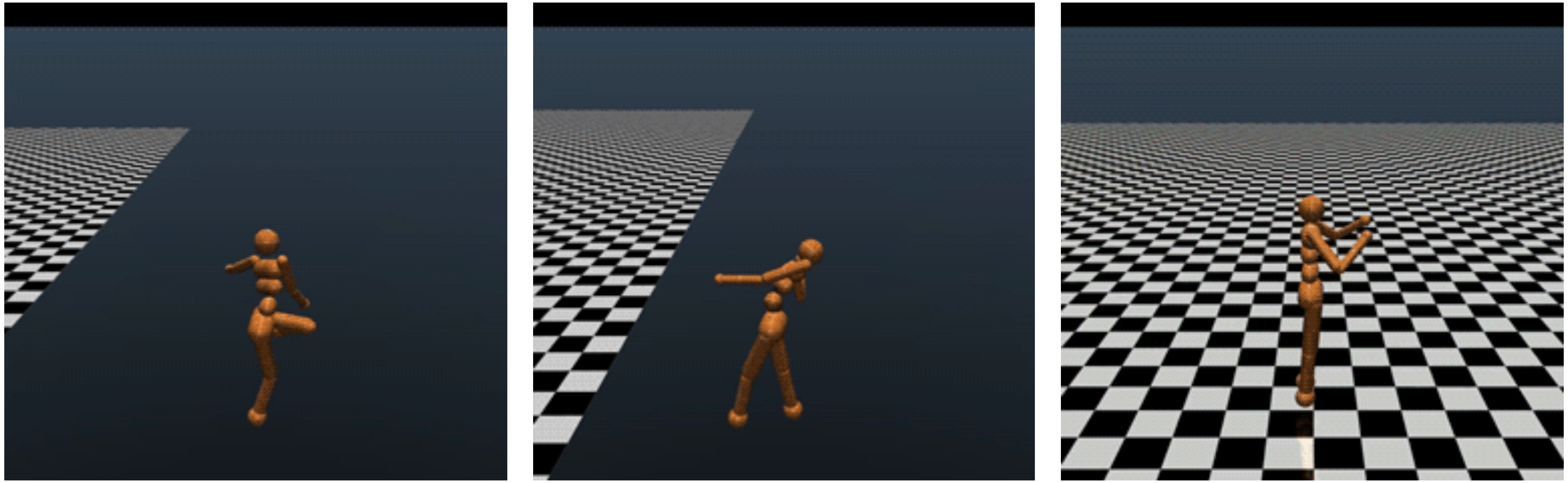
Train a robot to "run" forward as fast as possible

State: joint angles, center of mass, velocity, etc

Action: torques on joints

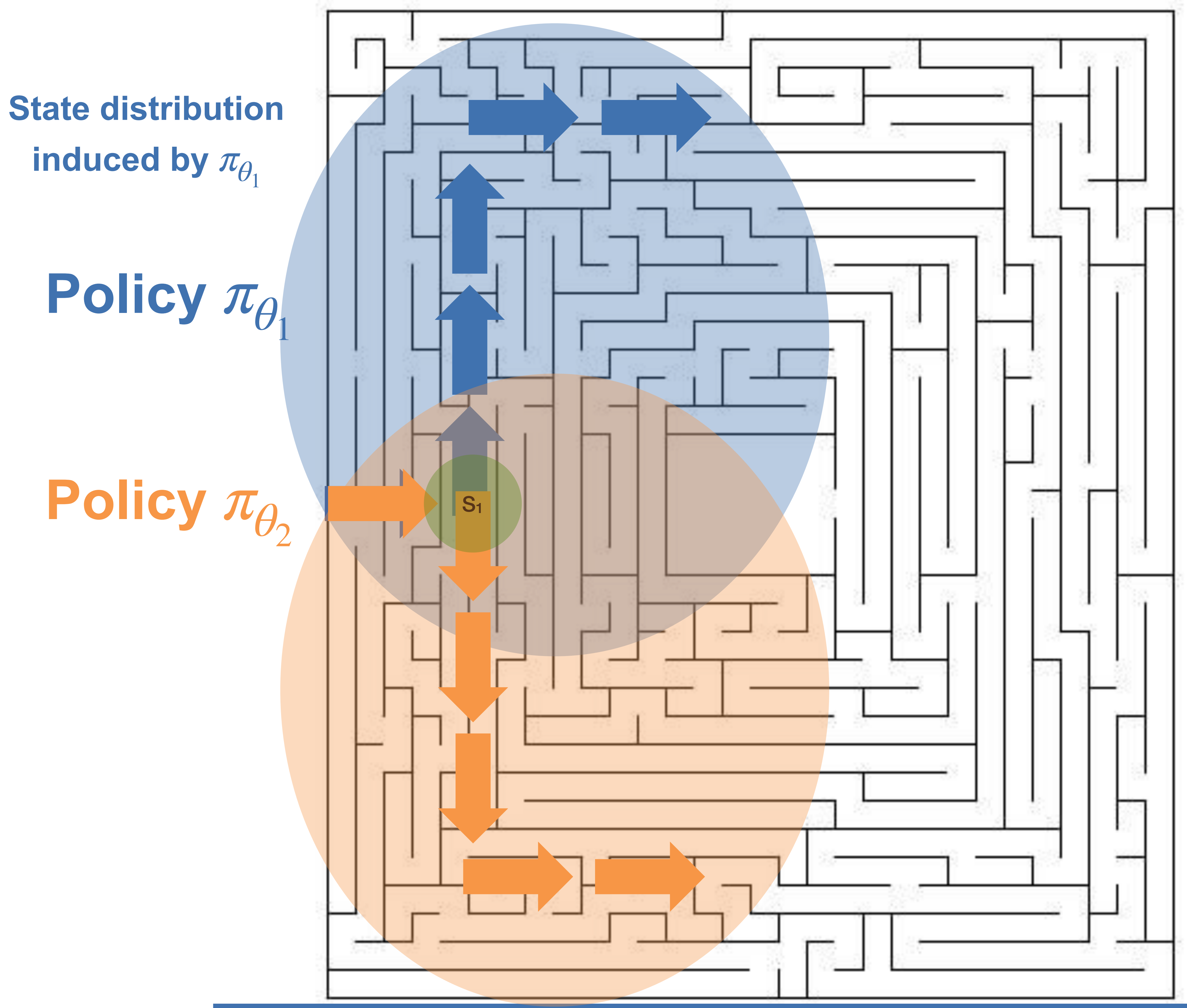
Reward: distance of moving forward between two steps

Note: All three robots achieve high reward!



Recall: $\nabla_\theta J(\pi_{\theta_t}) = \mathbb{E}_{s,a \sim d_\mu^{\pi_{\theta_t}}} \left[\nabla_\theta \ln \pi_{\theta_t}(a \mid s) A^{\pi_{\theta_t}}(s,a) \right]$

Policy: direction to move in at S_1



$$\mathscr{A} = \{a_{left}, a_{right}\}$$

$$\pi(a \mid s; \theta) = \begin{cases} 1 & \textbf{if } a = \arg\max f_\theta(a) \\ 0 & \textbf{otherwise} \end{cases}$$

$\theta_1 = (0.51, 0.49)$ [Parameter space]

$\pi_{\theta_1} : a_{left}$ [Policy space]

$$\theta_2 \leftarrow \theta_1 + \eta \nabla_\theta J(\pi_{\theta_1})$$
$$\leftarrow (0.51, 0.49) + (-0.02, ,0.02)$$
$$\leftarrow (0.49, 0.51)$$

$\pi_{\theta_2} : a_{right}$

Observation: Small changes in policy's parameters can lead to *large changes in policy*

Intuition Behind Observation #2

Observation 2: Small changes in policy's parameters can lead to *large changes in policy*

In other words…

"I don't care how big the change is to parameters (θ),
I care about the change to the policy (π_θ)"

Implicitly, PG considers Euclidean distance in **parameter space**

Our goal is to consider information from **policy space**

Intuition Behind Observation #2

Observation 2: Small changes in policy's parameters can lead to *large changes in policy*

Goal of New Approach

Perform **policy optimization** while considering "**policy change**"

Q: How do we measure "policy change"?

Goal of New Approach

Perform **policy optimization**
while considering “**policy change”**

Q: How do we measure “policy change”?

A: Look at trajectory distribution

$$\tau = \{s_0, a_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1}\}$$

$$\rho_\theta(\tau) = \mu(s_0)\pi_\theta(a_0 \mid s_0)P(s_1 \mid s_0, a_0)\pi_\theta(a_1 \mid s_1)\ldots$$

Goal of New Approach

Perform **policy optimization**
while considering "**policy change**"

At iteration t, with π_{θ_t} at hand, we compute θ_{t+1} as follows:

$$\max_{\pi_\theta} \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}} \left[\mathbb{E}_{a\sim\pi_\theta(s)} A^{\pi_{\theta_t}}(s,a) \right]$$

$$\text{s.t., } D_{KL}\left(\rho_{\pi_{\theta_t}} | \rho_{\pi_\theta}\right) \leq \delta$$

Q: What is D_{KL}?

We want to maximize local advantage against π_{θ_t},

but we want the new policy to be "close" to π_{θ_t}

KL-divergence: measures the distance between two distributions

Q: What is D_{KL}?

Given two distributions P & Q, where $P \in \Delta(X), Q \in \Delta(X)$,
KL Divergence is defined as:

$$KL(P \mid Q) = \mathbb{E}_{x \sim P}\left[\ln \frac{P(x)}{Q(x)}\right]$$

Examples:

If $Q = P$, then $KL(P \mid Q) = KL(Q \mid P) = 0$

If $P = \mathcal{N}(\mu_1, \sigma^2 I), Q = \mathcal{N}(\mu_2, \sigma^2 I)$, then $KL(P \mid Q) = \|\mu_1 - \mu_2\|_2^2 / \sigma^2$

Fact:

$KL(P \mid Q) \geq 0$, and being 0 if and only if $P = Q$

Outlines

1. Motivation behind trust-region policy optimization

2. Quick intro on KL-divergence

3. A Trust-Region Formulation for Policy Optimization

4. Algorithm: Natural Policy Gradient

A trust region formulation for policy update:

At iteration t, with π_{θ_t} at hand, we compute θ_{t+1} as follows:

$$\max_{\pi_\theta} \mathbb{E}_{s \sim d_\mu^{\pi_{\theta_t}}} \left[\mathbb{E}_{a \sim \pi_\theta(s)} A^{\pi_{\theta_t}}(s, a) \right]$$

$$\text{s.t.,}\ KL\left(\rho_{\pi_{\theta_t}} | \rho_{\pi_\theta}\right) \leq \delta$$

We want to maximize local advantage against π_{θ_t},

but want the new policy to be close to π_{θ_t}

Q: How do we compute KL between trajectory likelihoods?

A trust region formulation for policy update:

At iteration t, with π_{θ_t} at hand, we compute θ_{t+1} as follows:

$$\max_{\pi_\theta} \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}} \left[\mathbb{E}_{a\sim\pi_\theta(s)} A^{\pi_{\theta_t}}(s,a) \right]$$

$$\text{s.t., } KL\left(\rho_{\pi_{\theta_t}} | \rho_{\pi_\theta}\right) \leq \delta$$

Q: How do we compute KL between trajectory likelihoods?

High-level strategy

1. Simplify KL expression
2. Use Taylor expansion on KL expression

1. Simplifying KL constraint

Change from trajectory distribution to state-action distribution:

$$
\begin{aligned}
KL\left(\rho_{\pi_{\theta_t}} | \rho_{\pi_\theta}\right) &= \mathbb{E}_{\tau \sim \rho_{\pi_{\theta_t}}} \ln \frac{\rho_{\pi_{\theta_t}}(\tau)}{\rho_{\pi_\theta}(\tau)} \\
&= \mathbb{E}_{\tau \sim \rho_{\pi_{\theta_t}}} \sum_{h=0}^{H-1} \ln \frac{\pi_{\theta_t}(a_h | s_h)}{\pi_\theta(a_h | s_h)} \\
&= H \, \mathbb{E}_{s_h, a_h \sim d_\mu^{\pi_{\theta_t}}} \left[\ln \frac{\pi_{\theta_t}(a_h | s_h)}{\pi_\theta(a_h | s_h)} \right] \\
&:= \ell(\theta)
\end{aligned}
$$

Q: How do we approximate $\ell(\theta)$?

A: Taylor expansion

Recall: A trust region formulation for policy update:

At iteration t, with π_{θ_t} at hand, we compute θ_{t+1} as follows:

$$\max_{\pi_\theta} \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_\theta(s)} A^{\pi_{\theta_t}}(s,a)\right]$$

$$\text{s.t., } KL\left(\rho_{\pi_{\theta_t}} | \rho_{\pi_\theta}\right) \leq \delta$$

Q: How do we compute KL between trajectory likelihoods?

High-level strategy

✔ 1. Simplify KL

2. Use Taylor expansion on KL

2. Taylor expansion on KL

Gradients of KL

$$\ell(\theta_t) = 0$$

Recall $\ell(\theta) := H\, \mathbb{E}_{s,a\sim d^{\pi_{\theta_t}}}\left[\ln \frac{\pi_{\theta_t}(a\,|\,s)}{\pi_\theta(a\,|\,s)}\right]$

$$\nabla_\theta \ell(\theta) = 0\,|_{\theta=\theta_t}$$

$$\nabla_\theta^2 \ell(\theta) = \mathbb{E}\left[\nabla_\theta \ln \pi_{\theta_t}(a \mid s)\,\nabla_\theta \ln \pi_{\theta_t}(a \mid s)^\top\right]$$

Fisher Information Matrix $F(\theta_t)$

2. Taylor expansion on KL

Gradients of KL

$$\ell(\theta_t) = 0$$

$$\nabla_\theta \ell(\theta) = 0 \,\big|_{\theta=\theta_t}$$

$$\nabla_\theta^2 \ell(\theta) = F(\theta_t) = \mathbb{E}\left[\nabla_\theta \ln \pi_{\theta_t}(a \mid s)\, \nabla_\theta \ln \pi_{\theta_t}(a \mid s)^\top\right]$$

Taylor Expansion

$$\frac{1}{H} KL\left(\rho_{\pi_{\theta_t}} | \rho_{\pi_\theta}\right) = \ell(\theta)$$

$$\approx \frac{1}{2}(\theta - \theta_t)^\top F_{\theta_t}(\theta - \theta_t)$$

Outlines

1. Motivation behind trust-region policy optimization

2. Quick intro on KL-divergence

3. A Trust-Region Formulation for Policy Optimization

4. Algorithm: Natural Policy Gradient

Recall we have

At iteration t, we update to θ_{t+1} via:

$$\max_{\pi_\theta} \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_\theta(s)} A^{\pi_{\theta_t}}(s,a)\right]$$

$$\text{s.t.,}\ KL\left(\rho_{\pi_{\theta_t}} | \rho_{\pi_\theta}\right) \leq \delta$$

Simplify Objective Function

$$\max_{\theta} \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_\theta(s)} A^{\pi_{\theta_t}}(s,a)\right]$$

Since the objective is also non-linear,
let's do first order-talyor expansion on it:

$$\mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_\theta(s)} A^{\pi_{\theta_t}}(s,a)\right] \approx \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_{\theta_t}(s)} A^{\pi_{\theta_t}}(s,a)\right] + \underbrace{\mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_{\theta_t}(s)} \nabla_\theta \ln \pi_{\theta_t}(a\,|\,s) A^{\pi_{\theta_t}}(s,a)\right]}_{\nabla_\theta J(\pi_{\theta_t})} \cdot (\theta - \theta_t)$$

$$= \nabla_\theta J(\pi_{\theta_t})^\top (\theta - \theta_t)$$

Put everything together, we get:

At iteration t, we update to θ_{t+1} via:

Gradient update $$\max_{\theta} \nabla_\theta J(\pi_{\theta_t})^\top (\theta - \theta_t)$$

KL constraint $$\text{s.t. } (\theta - \theta_t)^\top F_{\theta_t} (\theta - \theta_t) \leq \delta$$

Linear objective and quadratic convex constraint: we can solve it optimally!

$$\theta_{t+1} = \theta_t + \eta F_{\theta_t}^{-1} \nabla_\theta J(\pi_{\theta_t})$$

Algorithm: Natural Policy Gradient

Initialize θ_0

For t = 0, …

Estimate PG $\nabla_\theta J(\pi_{\theta_t})$

Estimate Fisher info-matrix $F_{\theta_t} := \mathbb{E}_{s,a \sim d_\mu^{\pi_{\theta_t}}} \nabla_\theta \ln \pi_{\theta_t}(a \mid s) (\nabla_\theta \ln \pi_{\theta_t}(a \mid s))^\top$

Natural Gradient Ascent: $\theta_{t+1} = \theta_t + \eta F_{\theta_t}^{-1} \nabla_\theta J(\pi_{\theta_t})$

Summary

1. Motivation behind trust-region policy optimization

How can we optimize the policy's parameters while considering policy change?

2. Quick intro on KL-divergence

$$KL(P \mid Q) = \mathbb{E}_{x \sim P}\left[\ln \frac{P(x)}{Q(x)}\right]$$

3. A Trust-Region Formulation for Policy Optimization

$$\max_{\pi_\theta} \mathbb{E}_{s \sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a \sim \pi_\theta(s)} A^{\pi_{\theta_t}}(s, a)\right]$$

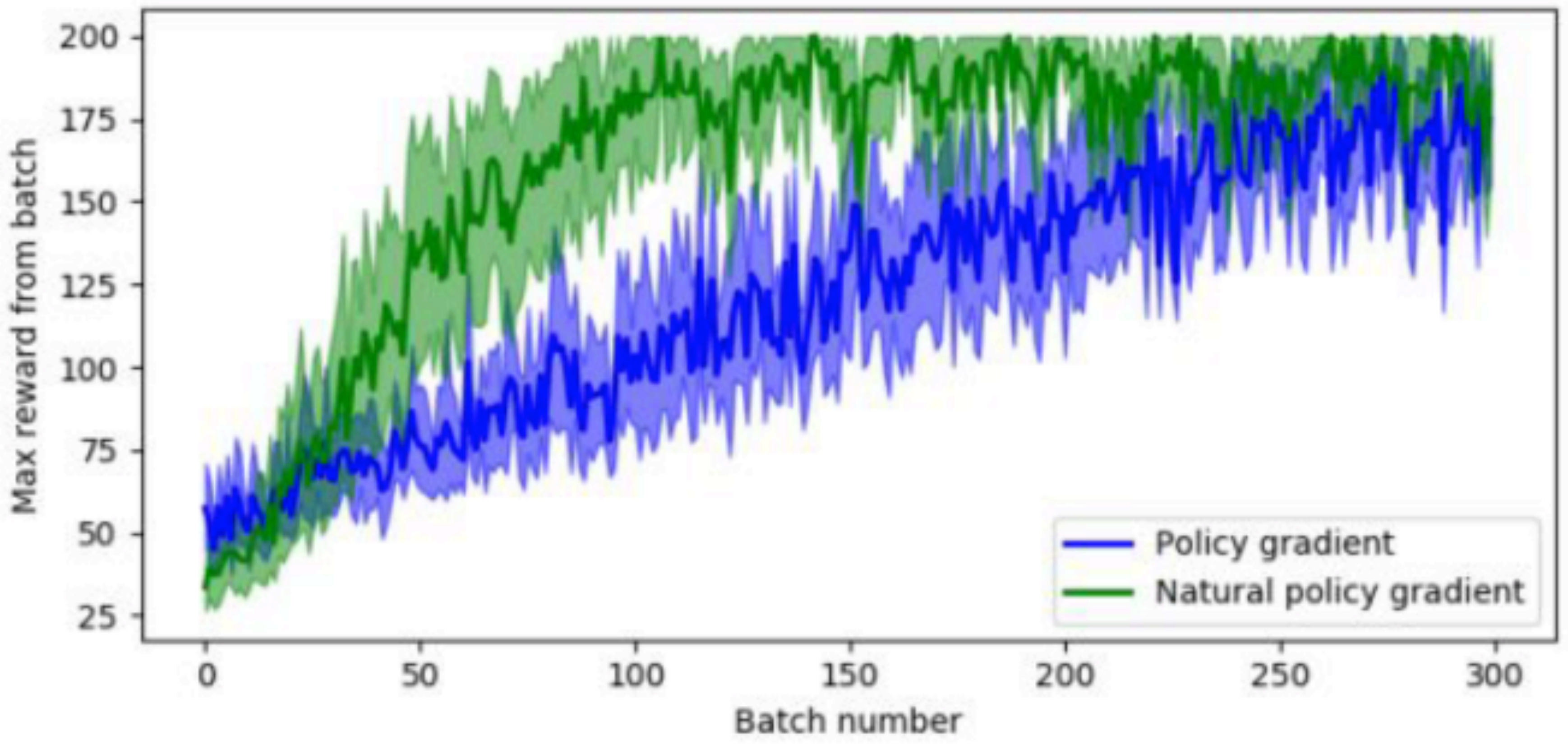
$$\text{s.t., } KL\left(\rho_{\pi_{\theta_t}} \mid \rho_{\pi_\theta}\right) \leq \delta$$

4. Algorithm: Natural Policy Gradient

$$\theta_{t+1} = \theta_t + \eta F_{\theta_t}^{-1} \nabla_\theta J(\pi_{\theta_t})$$

Fisher info-matrix $F_{\theta_t} := \mathbb{E}_{s,a \sim d_\mu^{\pi_{\theta_t}}} \nabla_\theta \ln \pi_{\theta_t}(a \mid s)(\nabla_\theta \ln \pi_{\theta_t}(a \mid s))^\top$

PG vs Natural PG



Review on Policy Optimization:

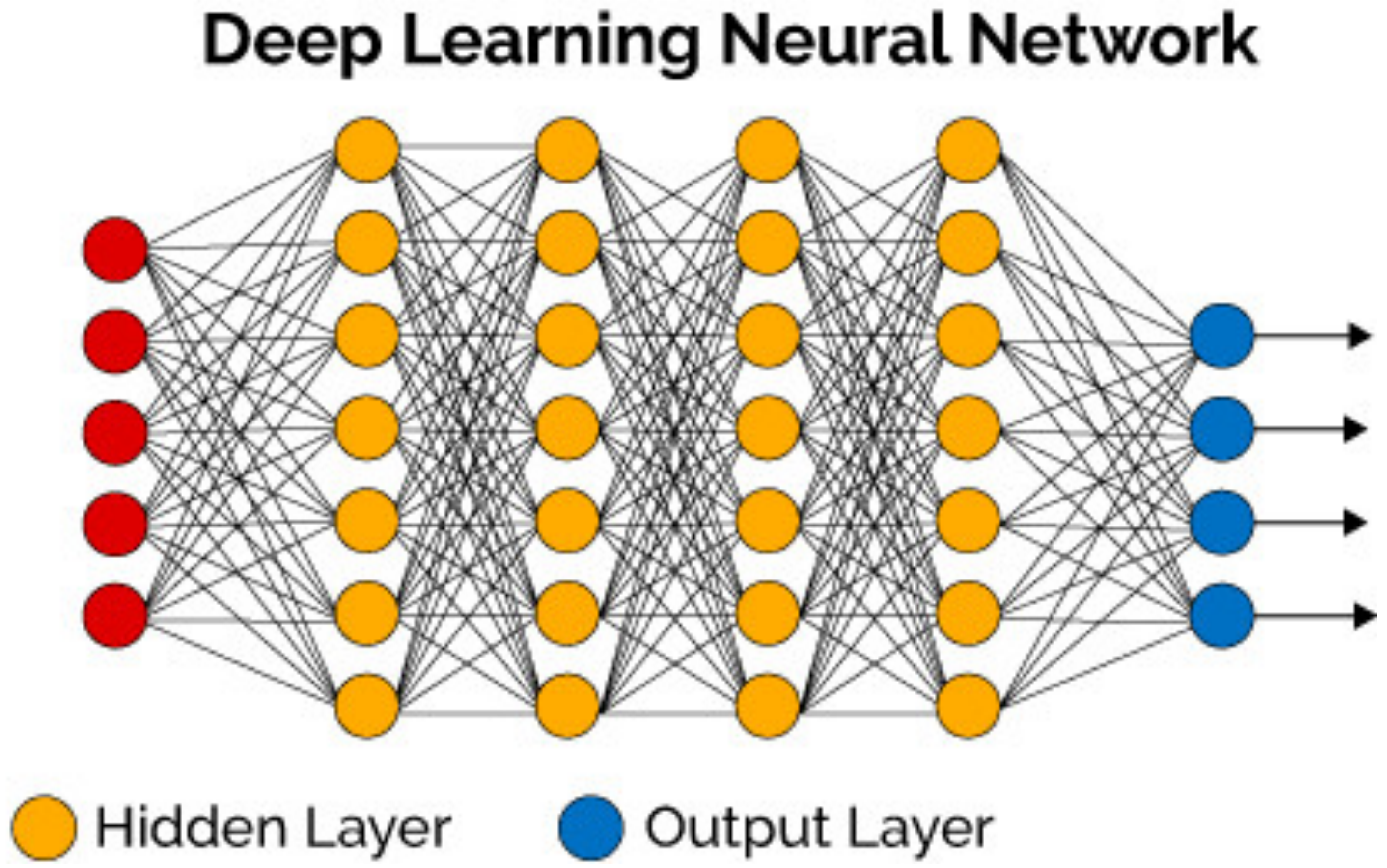
We have huge space space, i.e., $|S|$ might be $255^{3\times512\times512}$

We can only reset from initial state distribution $s_0 \sim \mu$

Numeration over state (e.g., a for loop) is not possible!

Goal: learn w/ function approximation

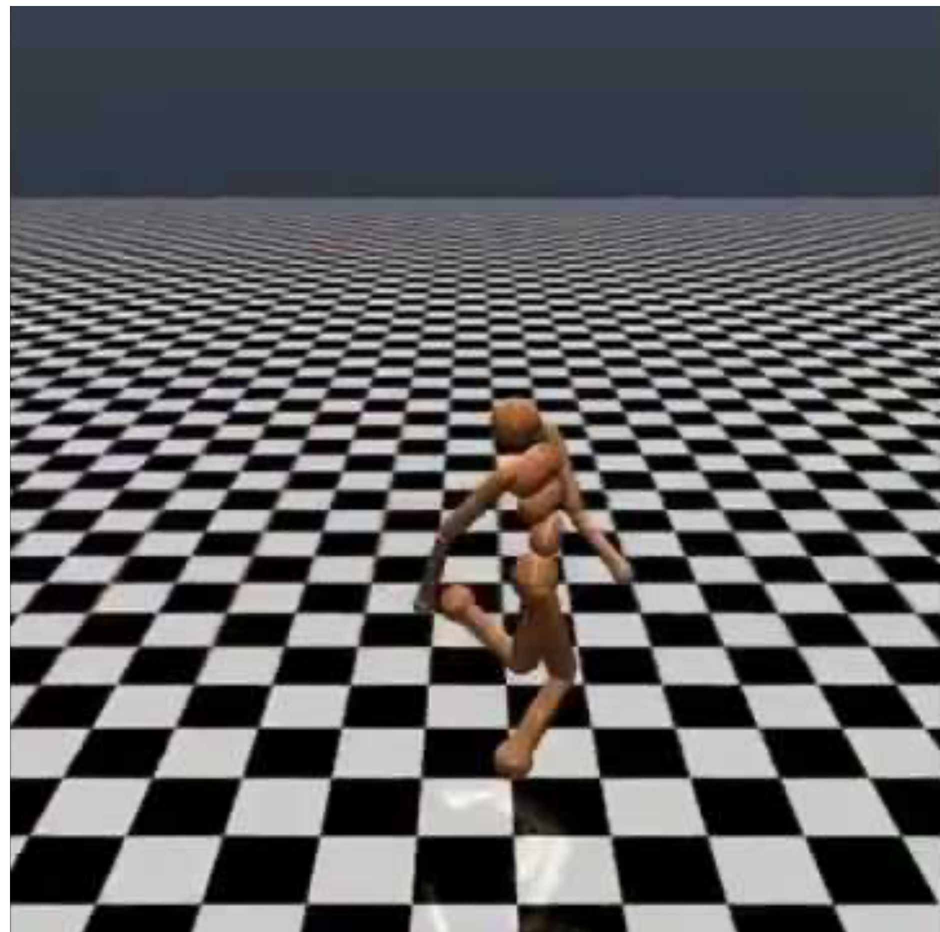
A Policy is a classifier w/ A many classes



What about continuous actions $a \in \mathbb{R}^d$?

$$\pi_{\beta,\alpha}(\cdot \mid s) = \mathcal{N}\left(\mu_\beta(s), \exp(\alpha) I_{d\times d}\right)$$

$$\theta := [\beta, \alpha]$$



Review on Policy Optimization: PG

Given an current policy π^t, we perform policy update to π^{t+1}

Third attempt: **PG on parameterized policy**

$$\max_{\theta} \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)} A^{\pi_{\theta_t}}(s,a)\right]$$

Locally Improve the local-adv a little bit via one-step gradient ascent:

$$\theta_{t+1} = \theta_t + \eta \cdot \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_{\theta_t}(s)} \nabla \ln \pi_{\theta_t}(a\,|\,s) \cdot A^{\pi_{\theta_t}}(s,a)\right]$$

When $\eta \to 0^+$, gradient ascent ensures
we improve the objective function

Review on Policy Optimization: NPG

Given an current policy π^t, we perform policy update to π^{t+1}

Fourth attempt: **Natural Policy Gradient**

$$\max_{\theta} \mathbb{E}_{s \sim d_\mu^{\pi_{\theta_t}}} \left[\mathbb{E}_{a \sim \pi_\theta(\cdot|s)} A^{\pi_{\theta_t}}(s,a) \right]$$

$$s.t., \mathrm{KL}(\rho_{\theta_t} | \rho_\theta) \leq \delta$$

Define fisher info-matrix $F_{\theta_t} = \nabla_\theta^2 \mathrm{KL}(\rho_{\theta_t} | \rho_\theta) |_{\theta=\theta_t}$,
a convex approximation, e.g., linearize obj and quadratize constraint,
gives us the following NPG update:

$$\max_{\theta} \nabla_\theta J(\pi_{\theta_t})^\top (\theta - \theta_t), \ \text{s.t.,} \ (\theta - \theta_t)^\top F_{\theta_t} (\theta - \theta_t) \leq \delta$$

An extension of NPG (even faster in practice):

Given an current policy π^t, we perform policy update to π^{t+1}

fifth attempt (new): **Proximal Policy Optimization (PPO)**

$$\max_\theta \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)} A^{\pi_{\theta_t}}(s,a)\right] \underbrace{-\lambda \mathbb{E}_{s\sim d_\mu^{\pi^t}}\left[\mathrm{KL}\left(\pi_{\theta_t}(a\,|\,s)\,|\,\pi_\theta(a\,|\,s)\right)\right]}_{\text{regularization}}$$

Use importance weighting & expand KL divergence:

$$\ell(\theta) := \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}}\left[\mathbb{E}_{a\sim\pi_{\theta_t}(\cdot|s)} \frac{\pi_\theta(a\,|\,s)}{\pi_{\theta_t}(a\,|\,s)} A^{\pi_{\theta_t}}(s,a)\right] - \lambda \mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}} \mathbb{E}_{a\sim\pi_{\theta_t}(\cdot|s)}\left[-\ln \pi_\theta(a\,|\,s)\right]$$

PPO: Perform a few steps of mini-batch SGA on $\ell(\theta)$ to approximate $\arg\max_\theta \ell(\theta)$