

CS 277: Control and Reinforcement Learning

Winter 2026

Lecture 6: Advanced Model-Free RL

Roy Fox

Department of Computer Science
School of Information and Computer Sciences
University of California, Irvine



Logistics

assignments

- Quiz 3 due next Monday
- Exercise 2 due next Friday

Today's lecture

Advantage estimation (cont.)

Convergence of RL

Continuous action spaces

Trust-region methods

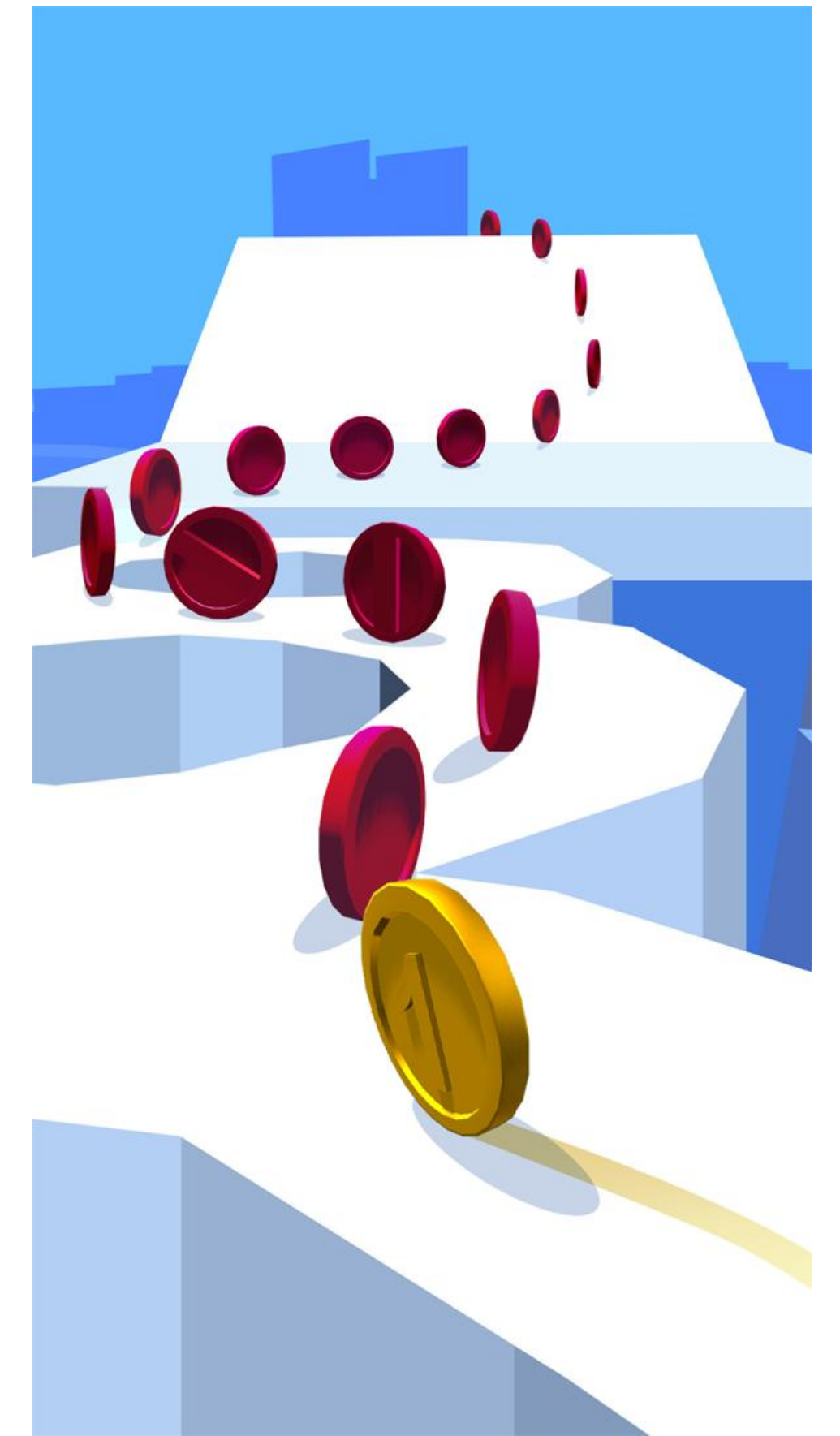
Comparing advantage estimators

	bias	variance
<ul style="list-style-type: none">Constant baseline $\nabla_{\theta} J_{\theta} \approx (R_{\geq t} - b) \nabla_{\theta} \log \pi_{\theta}(a_t s_t)$	none	high one gradient per trajectory
<ul style="list-style-type: none">State-based baseline (MC) $\nabla_{\theta} J_{\theta} \approx (R_{\geq t} - V_{\phi}(s_t)) \nabla_{\theta} \log \pi_{\theta}(a_t s_t)$	none	mid state-dependent baseline
<ul style="list-style-type: none">State-based baseline (TD) $\nabla_{\theta} J_{\theta} \approx (r_t + \gamma V_{\phi}(s_{t+1}) - V_{\phi}(s_t)) \nabla_{\theta} \log \pi_{\theta}(a_t s_t)$	some	lower

V_{ϕ} is approximate

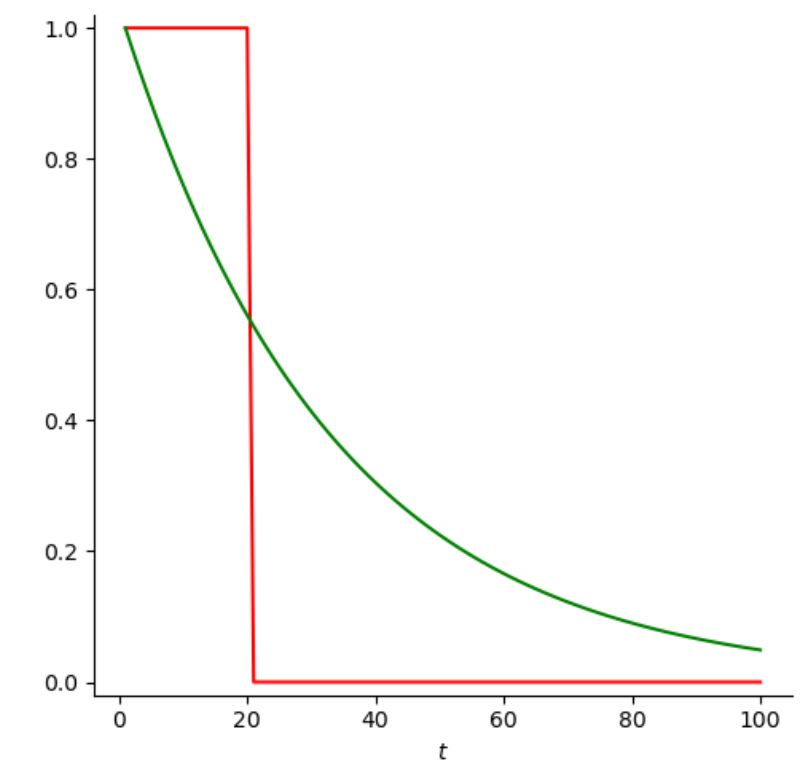
Multi-step TD

- 1-step TD: $A_t^1 = r_t + \gamma V(s_{t+1}) - V(s_t)$
- 2-step TD: $A_t^2 = r_t + \gamma r_{t+1} + \gamma^2 V(s_{t+2}) - V(s_t)$
- ...
- n -step TD: $A_t^n = r_t + \dots + \gamma^{n-1} r_{t+n-1} + \gamma^n V(s_{t+n}) - V(s_t)$
- In the limit (MC): $A_t^\infty = -V(s_t) + r_t + \gamma r_{t+1} + \dots$



TD(λ)

- How to choose n ?
 - Any specific n is **hard** truncation of the window of evidence we consider
- Instead, use **exponential window**
 - Take n -step TD with weight proportional to λ^n , where $0 \leq \lambda \leq 1$



$$A_t^\lambda = (1 - \lambda) \sum_n \lambda^{n-1} A_t^n = \sum_{\Delta t} (\lambda \gamma)^{\Delta t} (r_{t+\Delta t} + \gamma V(s_{t+\Delta t+1}) - V(s_{t+\Delta t}))$$

$A_{t+\Delta t}^1$

- **Generalized Advantage Estimation** (GAE(λ)): $\nabla_\theta J_\theta \approx A_t^\lambda \nabla_\theta \log \pi_\theta(a_t | s_t)$
 - GAE(1) = MC; GAE(0) = 1-step

Recap

- **Policy Gradient** = take the gradient of our objective w.r.t. policy parameters
 - Model-free, but **on-policy** and **high variance**
- **Variance reduction**:
 - **Past rewards** are independent of future actions
 - **TD** value estimation
 - **Baselines**, possibly state-dependent
 - **TD(λ)** to trade off bias and variance

Today's lecture

Advantage estimation (cont.)

Convergence of RL

Continuous action spaces

Trust-region methods

Backup operator

- **Value recursion:** $V_{\pi}(s) = \mathbb{E}_{(a|s) \sim \pi}[r(s, a) + \gamma \mathbb{E}_{(s'|s, a) \sim p}[V_{\pi}(s')]] = \mathcal{T}_{\pi}[V_{\pi}](s)$

linear backup operator

- In matrix notation:

$$\vec{v}_{\pi} = \vec{r}_{\pi} + \gamma P_{\pi} \vec{v}_{\pi}$$

$$\begin{bmatrix} \vec{r} \\ |\mathcal{S}| \end{bmatrix} + \gamma \begin{bmatrix} P \\ |\mathcal{S}| \times |\mathcal{S}| \end{bmatrix} \cdot \begin{bmatrix} \vec{v} \\ |\mathcal{S}| \end{bmatrix}$$

- ▶ where $r_{\pi}(s) = \mathbb{E}_{(a|s) \sim \pi}[r(s, a)]$, $P_{\pi}(s, s') = \mathbb{E}_{(a|s) \sim \pi}[p(s' | s, a)]$

$$\begin{matrix} & s' \\ s & \begin{bmatrix} P \\ |\mathcal{S}| \times |\mathcal{S}| \end{bmatrix} \end{matrix}$$

- Can be solved with **linear** algebra:

$$\vec{v}_{\pi} = (I - \gamma P_{\pi})^{-1} \vec{r}_{\pi}$$

largest eigenvalue magnitude

- The inverse always exists for $\gamma < 1$ because P_{π} has **spectral radius** 1

MF

θ

DP

π'

max

Bellman operator

MF

θ

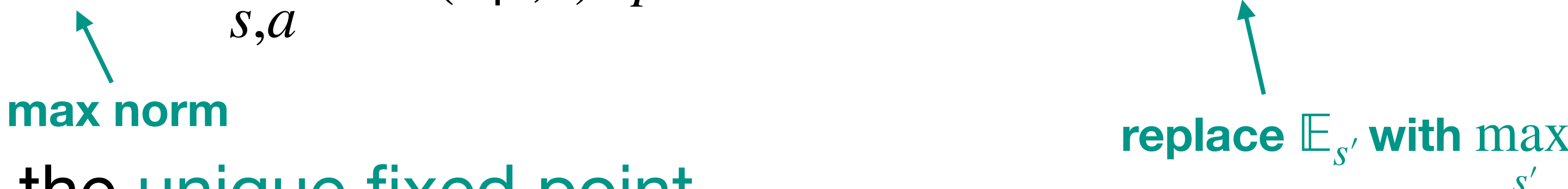
DP

π'

max

- **Bellman operator:** $\mathcal{T}[V](s) = \max_a r(s, a) + \gamma \mathbb{E}_{(s'|s,a) \sim p}[V(s')]$
 - Action-value version: $\mathcal{T}[Q](s, a) = r(s, a) + \gamma \mathbb{E}_{(s'|s,a) \sim p}[\max_{a'} Q(s', a')]$
- **Value Iteration** = iteratively apply \mathcal{T}
- Why is this guaranteed to converge? \mathcal{T} is a **contraction**:

$$\|\mathcal{T}[V_1] - \mathcal{T}[V_2]\|_{\infty} \leq \max_{s,a} \gamma \mathbb{E}_{(s'|s,a) \sim p}[V_1(s') - V_2(s')] \leq \gamma \|V_1(s') - V_2(s')\|_{\infty}$$



- $V^* = \mathcal{T}[V^*]$ is the **unique fixed point**

Q-Learning convergence

- **Q-Learning:** $Q(s, a) \rightarrow_{\alpha} r + \gamma \max_{a'} Q(s', a')$
 - ▶ In iteration i , use **learning rate** α_i
- **Robbins–Monro:** converges to Q^* with probability 1 (**almost surely**) if:
 - ▶ $\sum_i \alpha_i^2 < \infty$, implying $\alpha_i \rightarrow 0$ faster than $i^{-1/2}$
 - ▶ $\sum_i \alpha_i = \infty$, implying $\alpha_i \rightarrow 0$ not faster than i^{-1}
- **Example:** $\alpha_i = i^{-1}$ (like in averaging)

MF

θ

DP

π'

max

Fitted Value Iteration

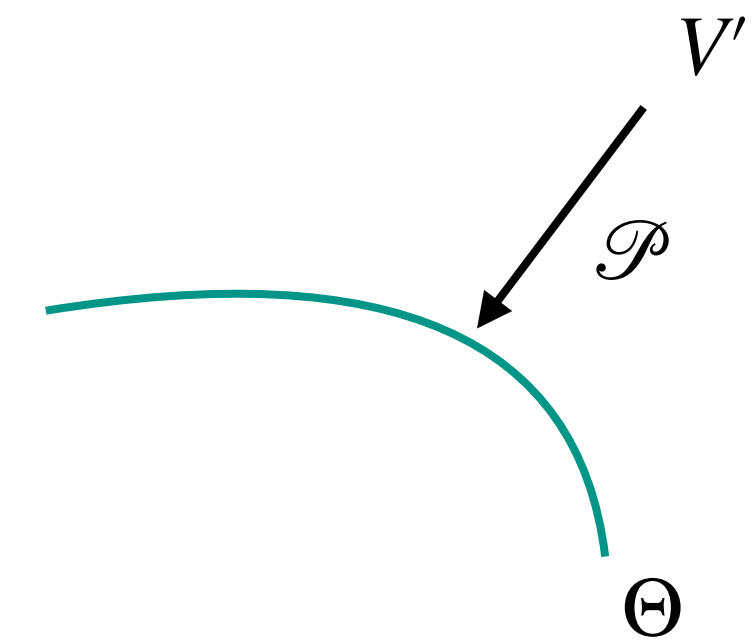
- Bellman (TD) error: $\mathcal{T}[V_{\bar{\theta}}](s) - V_{\theta}$
- Minimizing the square error is a **projection**

$$\mathcal{P}[V'] = V_{\arg \min_{\theta \in \Theta} \|V' - V_{\theta}\|_2^2}$$

- If Θ is convex, the projection is a **non-expansion**

$$\|\mathcal{P}[V'_1] - \mathcal{P}[V'_2]\|_2^2 \leq \|V'_1 - V'_2\|_2^2$$

- Composition of contractions contracts; but norms mismatch ($\mathcal{T}: L_{\infty}$; $\mathcal{P}: L_2$)
 - So $\mathcal{P}\mathcal{T}$ is generally not a contraction \Rightarrow **no convergence** guarantee for FVI



MF

θ

DP

π'

max

But isn't DQN just SGD?

Algorithm DQN

Initialize θ , set $\bar{\theta} \leftarrow \theta$

$s \leftarrow$ reset state

for each interaction step

Sample $a \sim \epsilon$ -greedy for $Q_{\theta}(s, \cdot)$

Get reward r and observe next state s'

Add (s, a, r, s') to replay buffer \mathcal{D}

Sample batch $(\vec{s}, \vec{a}, \vec{r}, \vec{s}') \sim \mathcal{D}$

$$y_i \leftarrow \begin{cases} r_i & s'_i \text{ terminal} \\ r_i + \gamma \max_{a'} Q_{\bar{\theta}}(s'_i, a') & \text{otherwise} \end{cases}$$

Descend $\mathcal{L}_{\theta} = (\vec{y} - Q_{\theta}(\vec{s}, \vec{a}))^2$

every T_{target} steps, set $\bar{\theta} \leftarrow \theta$

$s \leftarrow$ reset state if s' terminal, else $s \leftarrow s'$

MF

θ

DP

π'

max

moving target
≠ SGD

Is PG just SGD?

Algorithm REINFORCE

Initialize π_θ

repeat

Roll out $\xi \sim p_\theta$

Update with gradient $g \leftarrow R(\xi) \sum_t \nabla_\theta \log \pi_\theta(a_t | s_t)$

MF

θ

DP

π'

max

- The gradient is **unbiased** for $\nabla_\theta J_\theta$
- The objective J_θ changes with θ , but so does $\mathbb{E}_{x \sim D}[L_\theta(x)]$ in general ML
- But the **data distribution changes**
- Still, **convergence guaranteed** as long as we avoid $\pi(a | s) = 0$ [Agarwal et al., 2021]

Today's lecture

Advantage estimation (cont.)

Convergence of RL

Continuous action spaces

Trust-region methods

Continuous actions spaces

- What do we need for **policy-based** methods?

can do for large / continuous
action spaces?

- For rollouts: given s , **sample** from $\pi_{\theta}(a | s)$



- For policy update: given s and a , **compute** $\nabla_{\theta} \log \pi_{\theta}(a | s)$



- What do we need for **value-based** methods?

- For rollouts: given s , **compute** $\arg \max_a Q_{\theta}(s, a)$



- For value updates: given s , **compute** $\max_a Q_{\theta}(s, a)$



- How can we use **value-based** methods with continuous action spaces?

Idea 1: DQN with stochastic optimization

- Think of $\max_a Q(s, a)$ as an optimization problem, solved in inner loop
 - Example: **stochastic optimization** = learn ad-hoc approximately greedy policy π
- Run **value-based** algorithm; whenever it needs $\max_a Q(s, a)$ search for it:

Algorithm Stochastic optimization **a.k.a. Cross-Entropy Method (CRM)**

Initialize π

repeat

 Sample $a_1, \dots, a_k \sim \pi$

 Select k/c top values $Q(s, a_i)$ for $i = 1, \dots, k$

 Fit π to these “elites”

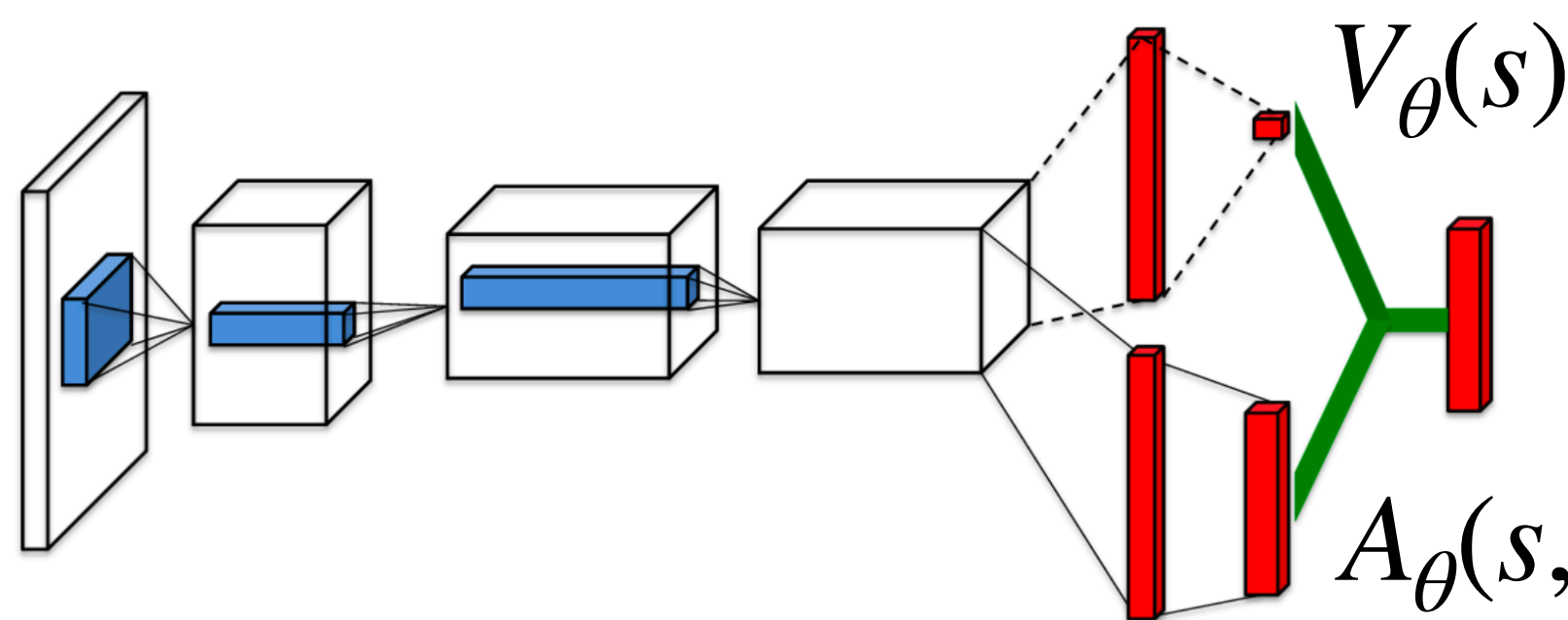
Idea 2: easily maximizable Q

- Represent Q_θ in a way that is directly **maximizable**
- Example: **quadratic** $Q_\theta(s, a) = -\frac{1}{2}(a - \mu_\theta(s))^T P_\theta(s)(a - \mu_\theta(s)) + V_\theta(s)$

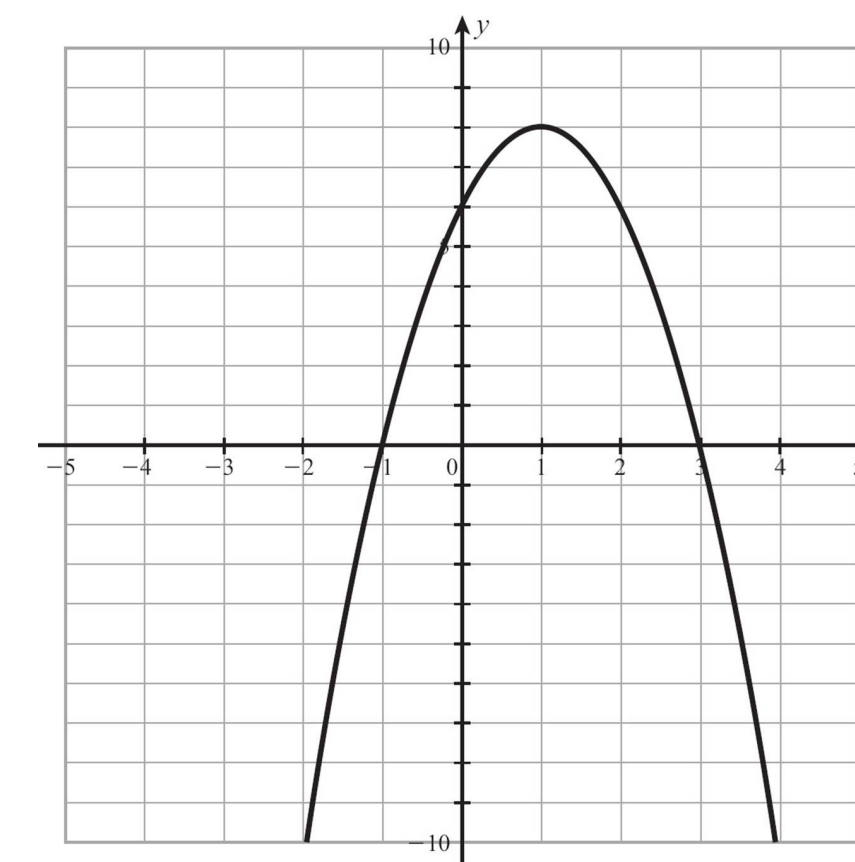
$$\arg \max_a Q_\theta(s, a) = \mu_\theta(s)$$

$$\max_a Q_\theta(s, a) = V_\theta(s)$$

- Possible architecture: **dueling network**



$$A_\theta(s, a) = -\frac{1}{2}(a - \mu_\theta(s))^T P_\theta(s)(a - \mu_\theta(s))$$



MF

θ

DP


π'

max


Idea 3: learn optimizing policy

- Previous methods: represent a Q maximizer or train one ad-hoc
- More general method: let a deterministic $\mu_\theta(s)$ learn to maximize $Q_\phi(s, a)$
 - This makes it an Actor–Critic method
- Deterministic Policy Gradient Theorem:

$$\begin{aligned}\nabla_\theta V_{\mu_\theta}(s) &= \nabla_\theta Q_{\mu_\theta}(s, \mu_\theta(s)) = \nabla_\theta Q_{\mu_{\bar{\theta}}}(s, \mu_\theta(s)) + \nabla_\theta Q_{\mu_\theta}(s, \mu_{\bar{\theta}}(s)) \\ &= \nabla_\theta Q_{\mu_{\bar{\theta}}}(s, \mu_\theta(s)) + \gamma \mathbb{E}_{(s'|s, \mu_\theta(s)) \sim p} [\nabla_\theta V_{\mu_\theta}(s')] \end{aligned}$$

pseudo-reward 

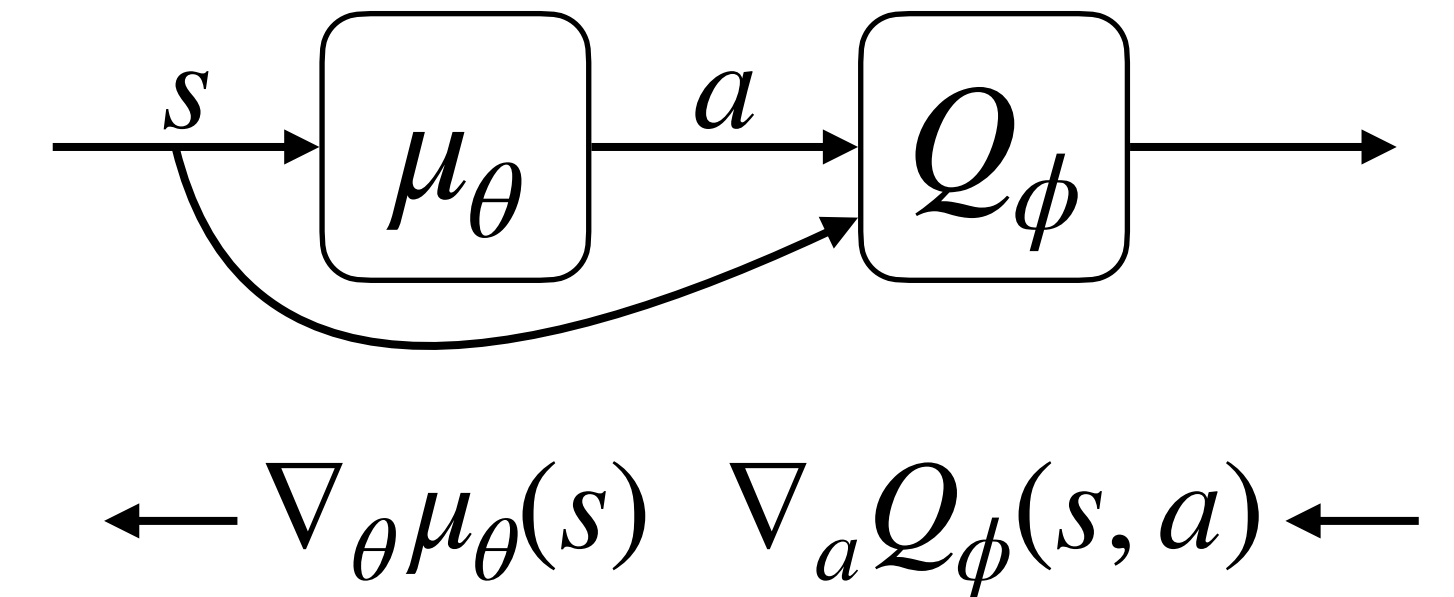
$$\nabla_\theta J_\theta = \sum_t \gamma^t \mathbb{E}_{s_t \sim p_\theta} [\nabla_\theta Q_{\mu_{\bar{\theta}}}(s_t, \mu_\theta(s_t))] = \frac{1}{1-\gamma} \mathbb{E}_{s \sim p_\theta} [\nabla_\theta Q_{\mu_{\bar{\theta}}}(s, \mu_\theta(s))]$$

$t \sim \text{Geo}(1 - \gamma)$ 

[Silver et al., 2014]

Deep Deterministic Policy Gradient (DDPG)

- Evaluating Q : feed actor $\mu_\theta(s)$ into critic $Q_\phi(s, a)$



- Back-propagation (chain rule):

$$\begin{aligned}\nabla_\theta J_\theta &= \mathbb{E}_{s \sim p_\theta} [\nabla_\theta Q_\phi(s, \mu_\theta(s))] \\ &= \mathbb{E}_{s \sim p_\theta} [\nabla_\theta \mu_\theta(s) \nabla_a Q_\phi(s, a = \mu_\theta(s))]\end{aligned}$$

- DDPG:

- Train critic Q_ϕ : TD policy evaluation
- Train actor π_θ : ascend $Q_\phi(s, \mu_\theta)$ with gradient through μ_θ

MF

θ

DP

π'

max

Today's lecture

Advantage estimation (cont.)

Convergence of RL

Continuous action spaces

Trust-region methods

Importance Sampling

- Suppose you want to estimate $\mathbb{E}_{x \sim p}[f(x)]$

- but only have samples $x \sim p'$

- Importance sampling:

$$\mathbb{E}_{x \sim p}[f(x)] = \mathbb{E}_{x \sim p'} \left[\frac{p(x)}{p'(x)} f(x) \right]$$

- Importance (IS) weights: $\rho(x) = \frac{p(x)}{p'(x)}$

- Estimate: $\rho(x)f(x)$ with $x \sim p'$



IS application 1: multi-step Q-Learning

- **n -step Q-Learning:** $Q(s_t, a_t) \rightarrow \sum_{\Delta t=0}^{n-1} \gamma^{\Delta t} r_{t+\Delta t} + \gamma^n \max_a Q(s_{t+n}, a)$
- Reminder: $Q^*(s_t, a_t)$ evaluates any a_t but optimal behavior afterward
 - ▶ We need data from $a_{t+\Delta t} = \arg \max_a Q(s_{t+\Delta t}, a)$ for RHS to estimate optimal target

- To be **off-policy**: update $Q(s_t, a_t) \rightarrow \sum_{\Delta t=0}^{n-1} \gamma^{\Delta t} \rho_t^{\Delta t} r_{t+\Delta t} + \gamma^n \max_a Q(s_{t+n}, a)$

- ▶ with $\rho_t^{\Delta t} = \prod_{i=t+1}^{t+\Delta t} \frac{\pi(a_i | s_i)}{\pi'(a_i | s_i)}$ for data from π'

MF

θ

DP

π'

max

IS application 2: off-policy policy evaluation

- Estimate $J_\pi = \mathbb{E}_{\xi \sim p_\pi}[R(\xi)]$ **off-policy**: $J_\pi = \mathbb{E}_{\xi \sim p_{\pi'}}[\rho_{\pi'}^\pi(\xi) R(\xi)]$

with $\rho_{\pi'}^\pi(\xi) = \frac{p_\pi(\xi)}{p_{\pi'}(\xi)} = \prod_t \frac{\pi(a_t | s_t)}{\pi'(a_t | s_t)}$ $\leftarrow p(s' | s, a)$ **cancels out**

- $\rho(\xi)$ can be very large or small \Rightarrow **high variance**
- Some **reduction**: r_t is not affected by future actions

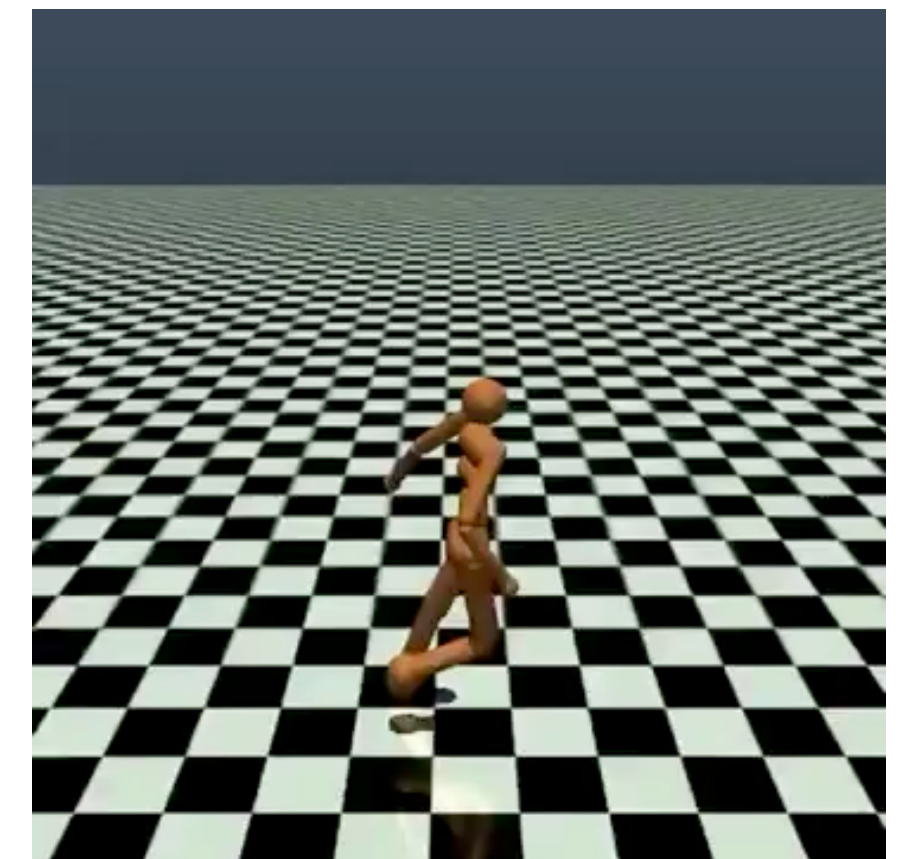
$$J_\pi = \sum_t \mathbb{E}_{\xi_{\leq t} \sim p_{\pi'}}[\gamma^t \rho_{\pi'}^\pi(\xi_{\leq t}) r_t] = \sum_t \mathbb{E}_{\xi_{\leq t} \sim p_{\pi'}} \left[\gamma^t r_t \prod_{t' \leq t} \frac{\pi(a_{t'} | s_{t'})}{\pi'(a_{t'} | s_{t'})} \right]$$

[Precup et al., 2000]

IS application 3: Off-policy Policy Gradient

- Policy Gradient: $\nabla_{\theta} J_{\theta} = \sum_t \gamma^t \mathbb{E}_{\xi \sim p_{\theta}} [R_{\geq t}(\xi) \nabla_{\theta} \log \pi_{\theta}(a_t | s_t)]$
- Off-Policy PG: $\nabla_{\theta} J_{\theta} = \sum_t \gamma^t \mathbb{E}_{\xi \sim p_{\theta'}} [\rho_{\theta'}^{\theta}(\xi_{\leq t}) R_{\geq t}(\xi) \nabla_{\theta} \log \pi_{\theta}(a_t | s_t)]$

- $R_{\geq t}(\xi)$ = future discounted rewards affected by $\pi_{\theta}(a_t | s_t)$
- $\rho_{\theta'}^{\theta}(\xi_{\leq t})$ = past probability ratios that affect $\pi_{\theta}(a_t | s_t)$



- Should we discount by γ^t ? Not if we care about evidence from later states

- $\rho_{\theta'}^{\theta}(\xi_{\leq t})$ has **high variance**, some methods just use $\rho_{\theta'}^{\theta}(a_t | s_t) = \frac{\pi_{\theta}(a_t | s_t)}{\pi_{\theta'}(a_t | s_t)}$

MF

θ

DP

π'

max

Performance Difference Lemma

- Policy gradient = small changes in policy; can we make large changes?

• For any π, ξ :
$$\sum_t \gamma^t A_\pi(s_t, a_t) = \sum_t \gamma^t (r_t + \gamma V_\pi(s_{t+1}) - V_\pi(s_t)) = R(\xi) - V_\pi(s_0)$$

telescopic cancellation
advantage of entire trajectory

- Expectation by different policy: Performance Difference Lemma

$$\sum_t \gamma^t \mathbb{E}_{(s_t, a_t) \sim p_\pi} [A_{\bar{\pi}}(s_t, a_t)] = \mathbb{E}_{\xi \sim p_\pi} [R(\xi) - V_{\bar{\pi}}(s_0)] = J_\pi - J_{\bar{\pi}}$$

$s_0 \sim p$ in both π and π'

- We want to maximize over π , with $\bar{\pi}$ fixed

• Compare: PG Theorem
$$\nabla_\theta J_\theta = \sum_t \gamma^t \mathbb{E}_{(s_t, a_t) \sim p_\theta} [A_{\pi_\theta}(s_t, a_t) \nabla_\theta \log \pi_\theta(a_t | s_t)]$$

Finding best next policy

- With **current policy** $\bar{\pi}$: find $\max_{\pi} J_{\pi} - J_{\bar{\pi}} = \max_{\pi} \sum_t \gamma^t \mathbb{E}_{(s_t, a_t) \sim p_{\pi}} [A_{\bar{\pi}}(s_t, a_t)]$
 - Can use $\bar{\pi}$ to **evaluate** $A_{\bar{\pi}}$
- But we don't have data $(s_t, a_t) \sim p_{\pi}$; **idea**: sample from $\bar{\pi}$
 - **Trick question**: is this on-policy or off-policy? **On-policy** data, but needs **IS weight**

$$\max_{\pi} \sum_t \gamma^t \mathbb{E}_{\xi_{\leq t} \sim p_{\bar{\pi}}} [\rho_{\bar{\pi}}^{\pi}(\xi_{\leq t}) A_{\bar{\pi}}(s_t, a_t)]$$

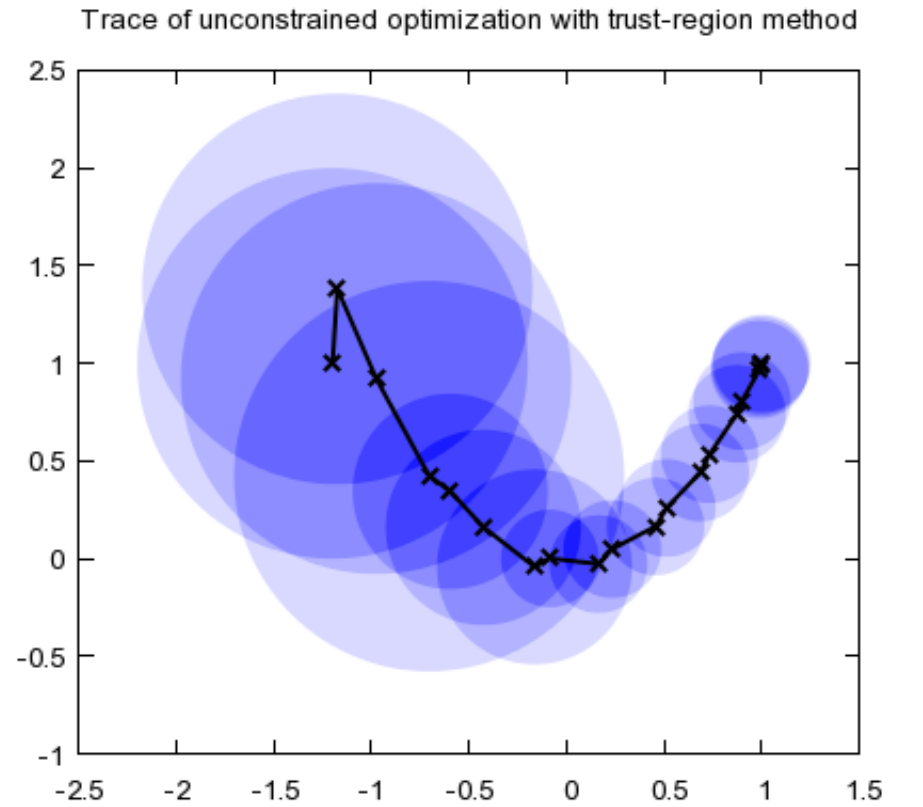
- Is it reasonable to use $\rho_{\bar{\pi}}^{\pi}(a_t | s_t) = \frac{\pi(a_t | s_t)}{\bar{\pi}(a_t | s_t)}$ instead? i.e. drop $\rho_{\bar{\pi}}^{\pi}(\xi_{<t})$

Trust-Region Policy Optimization (TRPO)

- **Trust region** = space around $\bar{\pi}$ where $\rho(\xi_{<t}) \approx 1$

- ▶ Easier to consider $\mathbb{E}_{\xi_{<t} \sim p_{\bar{\pi}}}[\log \rho(\xi_{<t})] \approx 0$

$$-\mathbb{E}_{\xi_{<t} \sim p_{\bar{\pi}}}[\log \rho(\xi_{<t})] = \mathbb{D}[\bar{\pi}(\xi_{<t}) \parallel \pi(\xi_{<t})] = \sum_{t' < t} \mathbb{E}_{\xi_{<t'} \sim p_{\bar{\pi}}}[\mathbb{D}[\bar{\pi}(a_{t'} \mid s_{t'}) \parallel \pi(a_{t'} \mid s_{t'})]]$$



- **TRPO**: $\max_{\theta} \mathbb{E}_{(s,a) \sim p_{\bar{\theta}}}[\rho_{\bar{\theta}}^{\theta}(a \mid s) A_{\bar{\theta}}(s, a)]$ s.t. $\mathbb{E}_{s \sim p_{\bar{\theta}}}[\mathbb{D}[\pi_{\bar{\theta}}(a \mid s) \parallel \pi_{\theta}(a \mid s)]] \leq \epsilon$

- ▶ $A_{\bar{\theta}}$ estimated with **critic** A_{ϕ}

- ▶ Computational tricks for **gradient-based optimization**

MF

θ

DP

π'

max

Proximal Policy Optimization (PPO)

- Same motivation: ascend $\mathbb{E}_{(s,a) \sim p_{\bar{\theta}}}[\rho_{\bar{\theta}}^{\theta}(a | s) A_{\bar{\theta}}(s, a)]$ with π_{θ} staying near $\pi_{\bar{\theta}}$
 - **PPO-Penalty**: add a penalty term for $\mathbb{E}_{s \sim p_{\bar{\theta}}}[\mathbb{D}[\pi_{\bar{\theta}}(a | s) || \pi_{\theta}(a | s)]]$
 - **PPO-Clip**: ascend $\mathbb{E}_{(s,a) \sim p_{\bar{\theta}}}[L_{\bar{\theta}}^{\theta}(s, a)]$ with

$$L_{\bar{\theta}}^{\theta}(s, a) = \min(\rho_{\bar{\theta}}^{\theta}(a | s) A_{\bar{\theta}}(s, a), A_{\bar{\theta}}(s, a) + |\epsilon A_{\bar{\theta}}(s, a)|)$$

- Positive / negative advantage \Rightarrow increase / decrease $\rho_{\bar{\theta}}^{\theta}(a | s) = \frac{\pi_{\theta}(a | s)}{\pi_{\bar{\theta}}(a | s)}$

- But **no incentive** beyond $\rho_{\bar{\theta}}^{\theta}(a | s) = 1 \pm \epsilon$

- **no incentive \neq doesn't happen**
- **PPO has lots more tricks to limit divergence**

MF

θ

DP

π'

max

Recap

- Model-based policy evaluation can be solved linearly
- Deep RL isn't just SGD
 - Exception: policy gradient on offline (batch) data
- Value-based methods struggle to max in continuous action spaces
 - DDPG: π_θ learns to maximize Q_ϕ (actor–critic method)
- Importance Sampling decouples expectation and sampling distributions
 - Optimize on-policy objectives with off-policy data
 - TRPO and PPO: sample from current policy to evaluate next policy, if it's close

Logistics

assignments

- Quiz 3 due next Monday
- Exercise 2 due next Friday