# Lecture 9: Policy Optimization

In the previous lectures, we studied value-based methods: value iteration, policy iteration, and policy evaluation with function approximation. These methods operate by estimating value functions and extracting policies indirectly. We now turn to a fundamentally different approach: *direct policy optimization*. Instead of learning a value function and deriving a policy from it, we parameterize a policy $\pi_\theta$ directly and optimize the objective $J(\theta) = V^{\pi_\theta}(\mu)$ via gradient-based methods.

Why optimize policies directly? Several motivations arise naturally:

- **Continuous/high-dimensional actions.** Value-based methods require $\operatorname{argmax}_a Q(s, a)$, which is intractable when $\mathcal{A}$ is continuous or large. Policy parameterizations avoid this by directly outputting actions.
- **Stochastic policies.** In partially observed environments or games, optimal behavior may require randomization. Policy parameterizations naturally represent stochastic policies.
- **Function approximation compatibility.** Gradient-based optimization of parameterized policies integrates seamlessly with neural network function approximation.
- **Smooth optimization landscape.** Small changes to $\theta$ produce small changes in $\pi_\theta$, enabling gradient-based optimization with well-understood convergence properties.

This lecture develops the policy gradient theorem and the REINFORCE algorithm, introduces variance reduction and actor-critic methods, and studies three approaches to conservative policy updates: Conservative Policy Iteration (CPI), Trust Region Policy Optimization (TRPO), and Proximal Policy Optimization (PPO). The policy evaluation methods from Lecture 8 will serve as the "critic" component in actor-critic algorithms.

## Policy Parameterization

Consider a class of parametric policies $\{\pi_\theta \mid \theta \in \Theta \subset \mathbb{R}^d\}$. For a distribution $\rho$ over states, define $V^\pi(\rho) \coloneqq \mathbb{E}_{s_0 \sim \rho}[V^\pi(s_0)]$. The optimization problem of interest is:

$$\max_{\theta \in \Theta} \; V^{\pi_\theta}(\rho).$$

One immediate issue is that if the policy class $\{\pi_\theta\}$ consists of deterministic policies, then $\pi_\theta$ will, in general, not be differentiable. This motivates stochastic policy classes that permit differentiation.

**Definition 1** (Softmax policy)**.** *The* tabular softmax policy *is given by*

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

*where the parameter space is* $\Theta = \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$.

Note that the closure of the set of softmax policies contains all stationary policies, including deterministic ones (achieved by sending parameters to $\pm\infty$).

**Definition 2** (Log-linear policy)**.** *Given a feature mapping* $\phi_{s,a} \in \mathbb{R}^d$ *for each state-action pair, the* log-linear policy *is*

$$\pi_\theta(a \mid s) = \frac{\exp(\theta^\mathsf{T} \phi_{s,a})}{\sum_{a' \in \mathcal{A}} \exp(\theta^\mathsf{T} \phi_{s,a'})},$$

*with* $\theta \in \mathbb{R}^d$.

More generally, for a *neural softmax policy*, one may parameterize $\pi_\theta(a \mid s) = \exp(f_\theta(s,a))/\sum_{a'} \exp(f_\theta(s,a'))$ where $f_\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a neural network.

The softmax parameterization represents all stochastic policies (and its closure includes deterministic policies). Since the parameters $\theta$ are unconstrained, standard optimization tools apply directly. The log-linear and neural variants allow generalization across states through shared parameters.

# The Policy Gradient Theorem

## Deriving the Policy Gradient

Throughout this lecture, $\nabla$ denotes the gradient with respect to the policy parameter $\theta$ unless otherwise noted.

Let $\tau$ denote a trajectory with distribution $\mathbb{P}_\mu^{\pi_\theta}(\tau) = \mu(s_0)\pi_\theta(a_0 \mid s_0)P(s_1 \mid s_0, a_0)\pi_\theta(a_1 \mid s_1)\cdots$ under starting distribution $\mu$. Define the discounted total reward:

$$R(\tau) \coloneqq \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t).$$

Observe that $V^{\pi_\theta}(\mu) = \mathbb{E}_{\tau \sim \mathbb{P}_\mu^{\pi_\theta}}[R(\tau)]$.

**Theorem 1** (Policy gradient theorem (Sutton et al., 1999))**.** *The following are expressions for* $\nabla V^{\pi_\theta}(\mu)$*:*

*(i)* ***REINFORCE form:***

$$\nabla V^{\pi_\theta}(\mu) = \mathbb{E}_{\tau\sim\mathbb{P}_\mu^{\pi_\theta}}\left[R(\tau)\sum_{t=0}^{\infty}\nabla\log\pi_\theta(a_t\mid s_t)\right].$$

*(ii)* ***Action-value form:***

$$\begin{aligned}\nabla V^{\pi_\theta}(\mu) &= \mathbb{E}_{\tau\sim\mathbb{P}_\mu^{\pi_\theta}}\left[\sum_{t=0}^{\infty}\gamma^t Q^{\pi_\theta}(s_t,a_t)\nabla\log\pi_\theta(a_t\mid s_t)\right]\\ &= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi_\theta}}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[Q^{\pi_\theta}(s,a)\nabla\log\pi_\theta(a\mid s)\right].\end{aligned}$$

*(iii)* ***Advantage form:***

$$\nabla V^{\pi_\theta}(\mu) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi_\theta}}\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[A^{\pi_\theta}(s,a)\nabla\log\pi_\theta(a\mid s)\right].$$

*Here $d^{\pi_\theta} = d_\mu^{\pi_\theta}$ denotes the discounted state visitation distribution under $\pi_\theta$ starting from $\mu$ (as defined in Lecture 1).*

**Policy Gradient Theorem.**

$$\begin{aligned}\nabla V^{\pi_\theta}(\mu) &= \mathbb{E}_{\tau\sim\mathbb{P}_\mu^{\pi_\theta}}\left[R(\tau)\sum_{t=0}^{\infty}\nabla\log\pi_\theta(a_t\mid s_t)\right] && \text{(REINFORCE)}\\ &= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi_\theta},a\sim\pi_\theta}\left[Q^{\pi_\theta}(s,a)\nabla\log\pi_\theta(a\mid s)\right] && \text{(Action-value)}\\ &= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi_\theta},a\sim\pi_\theta}\left[A^{\pi_\theta}(s,a)\nabla\log\pi_\theta(a\mid s)\right] && \text{(Advantage)}\end{aligned}$$

*Proof.* **(i) REINFORCE form.** We have:

$$\begin{aligned}\nabla V^{\pi_\theta}(\mu) &= \nabla\sum_\tau R(\tau)\mathbb{P}_\mu^{\pi_\theta}(\tau) = \sum_\tau R(\tau)\nabla\mathbb{P}_\mu^{\pi_\theta}(\tau)\\ &= \sum_\tau R(\tau)\mathbb{P}_\mu^{\pi_\theta}(\tau)\nabla\log\mathbb{P}_\mu^{\pi_\theta}(\tau)\\ &= \sum_\tau R(\tau)\mathbb{P}_\mu^{\pi_\theta}(\tau)\nabla\log\left(\mu(s_0)\pi_\theta(a_0\mid s_0)P(s_1\mid s_0,a_0)\pi_\theta(a_1\mid s_1)\cdots\right)\\ &= \sum_\tau R(\tau)\mathbb{P}_\mu^{\pi_\theta}(\tau)\left(\sum_{t=0}^{\infty}\nabla\log\pi_\theta(a_t\mid s_t)\right)\end{aligned}$$

$$= \mathbb{E}_{\tau\sim\mathbb{P}_\mu^{\pi_\theta}}\left[R(\tau)\sum_{t=0}^{\infty}\nabla\log\pi_\theta(a_t\mid s_t)\right].$$

The key step uses the "log-derivative trick": $\nabla\mathbb{P}(\tau) = \mathbb{P}(\tau)\nabla\log\mathbb{P}(\tau)$. The transition probabilities $P$ and initial distribution $\mu$ do not depend on $\theta$, so their gradients vanish.

**(ii) Action-value form.** For any state $s_0$:

$$\begin{aligned}
\nabla V^{\pi_\theta}(s_0) &= \nabla\sum_{a_0}\pi_\theta(a_0\mid s_0)Q^{\pi_\theta}(s_0,a_0)\\
&= \sum_{a_0}\Big(\nabla\pi_\theta(a_0\mid s_0)\Big)Q^{\pi_\theta}(s_0,a_0)+\sum_{a_0}\pi_\theta(a_0\mid s_0)\nabla Q^{\pi_\theta}(s_0,a_0)\\
&= \sum_{a_0}\pi_\theta(a_0\mid s_0)\Big(\nabla\log\pi_\theta(a_0\mid s_0)\Big)Q^{\pi_\theta}(s_0,a_0)\\
&\quad+\sum_{a_0}\pi_\theta(a_0\mid s_0)\nabla\Big(r(s_0,a_0)+\gamma\sum_{s_1}P(s_1\mid s_0,a_0)V^{\pi_\theta}(s_1)\Big)\\
&= \mathbb{E}_{\tau\sim\mathbb{P}_{s_0}^{\pi_\theta}}\left[Q^{\pi_\theta}(s_0,a_0)\nabla\log\pi_\theta(a_0\mid s_0)\right]+\gamma\mathbb{E}_{\tau\sim\mathbb{P}_{s_0}^{\pi_\theta}}\left[\nabla V^{\pi_\theta}(s_1)\right].
\end{aligned}$$

By linearity of expectation and recursion:

$$\begin{aligned}
\nabla V^{\pi_\theta}(\mu) &= \mathbb{E}_{\tau\sim\mathbb{P}_\mu^{\pi_\theta}}\left[Q^{\pi_\theta}(s_0,a_0)\nabla\log\pi_\theta(a_0\mid s_0)\right]+\gamma\mathbb{E}_{\tau\sim\mathbb{P}_\mu^{\pi_\theta}}\left[\nabla V^{\pi_\theta}(s_1)\right]\\
&= \mathbb{E}_{\tau\sim\mathbb{P}_\mu^{\pi_\theta}}\left[Q^{\pi_\theta}(s_0,a_0)\nabla\log\pi_\theta(a_0\mid s_0)\right]+\gamma\mathbb{E}_{\tau\sim\mathbb{P}_\mu^{\pi_\theta}}\left[Q^{\pi_\theta}(s_1,a_1)\nabla\log\pi_\theta(a_1\mid s_1)\right]+\cdots\\
&= \mathbb{E}_{\tau\sim\mathbb{P}_\mu^{\pi_\theta}}\left[\sum_{t=0}^{\infty}\gamma^t Q^{\pi_\theta}(s_t,a_t)\nabla\log\pi_\theta(a_t\mid s_t)\right] = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\pi_\theta},a\sim\pi_\theta}\left[Q^{\pi_\theta}(s,a)\nabla\log\pi_\theta(a\mid s)\right].
\end{aligned}$$

**(iii) Advantage form.** Since $\sum_a\pi_\theta(a\mid s)\nabla\log\pi_\theta(a\mid s) = \sum_a\nabla\pi_\theta(a\mid s) = \nabla 1 = 0$, we can subtract any state-dependent baseline from $Q^{\pi_\theta}$ without changing the gradient. Taking $V^{\pi_\theta}(s)$ as the baseline gives $A^{\pi_\theta}(s,a) = Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s)$. □

## Non-Concavity of the Objective

**Lemma 2** (Non-concavity (Agarwal et al., 2021, Lemma 9.5))**.** *There exists an MDP $M$ such that the optimization problem $V^{\pi_\theta}(s)$ is not concave in $\theta$ for the softmax parameterization.*

*Proof.* Consider the deterministic 5-state MDP in Figure 1. Actions in terminal states $s_3$, $s_4$, and $s_5$ do not change the expected reward, so we only consider actions in $s_1$ and $s_2$. Let action $a_1$ be "up/above" and $a_2$ be "right." Then:

$$V^{\pi}(s_1) = \pi(a_2\mid s_1)\cdot\pi(a_1\mid s_2)\cdot r.$$

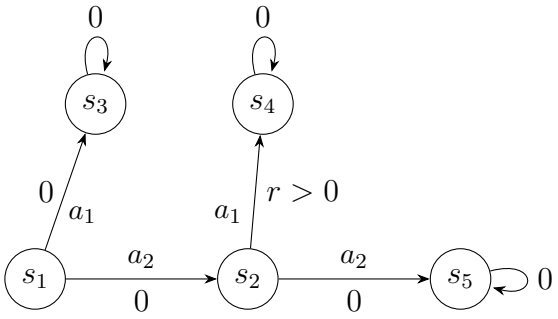



Figure 1: A deterministic MDP where $V^{\pi_\theta}(s)$ is not concave in $\theta$ for the softmax parameterization. Numbers on arrows represent rewards.

Write $\theta = (\theta_{a_1,s_1}, \theta_{a_2,s_1}, \theta_{a_1,s_2}, \theta_{a_2,s_2})$. Consider $\theta^{(1)} = (0, \log 3, \log 3, 0)$ and $\theta^{(2)} = (0, -\log 3, -\log 3, 0)$. By the softmax formula $\pi_\theta(a \mid s) = \exp(\theta_{a,s}) / \sum_{a'} \exp(\theta_{a',s})$:

$$\pi^{(1)}(a_2 \mid s_1) = \frac{\exp(\log 3)}{\exp(0) + \exp(\log 3)} = \frac{3}{1+3} = \frac{3}{4}, \quad \pi^{(1)}(a_1 \mid s_2) = \frac{3}{4},$$
$$\pi^{(2)}(a_2 \mid s_1) = \frac{\exp(-\log 3)}{\exp(0) + \exp(-\log 3)} = \frac{1/3}{1+1/3} = \frac{1}{4}, \quad \pi^{(2)}(a_1 \mid s_2) = \frac{1}{4}.$$

So $V^{(1)}(s_1) = \frac{3}{4} \cdot \frac{3}{4} \cdot r = \frac{9}{16} r$ and $V^{(2)}(s_1) = \frac{1}{4} \cdot \frac{1}{4} \cdot r = \frac{1}{16} r$.

For $\theta^{\text{mid}} = \frac{\theta^{(1)} + \theta^{(2)}}{2} = (0, 0, 0, 0)$:

$$\pi^{\text{mid}}(a \mid s) = \frac{\exp(0)}{\exp(0) + \exp(0)} = \frac{1}{2} \quad \text{for all } a, s,$$

so $V^{\text{mid}}(s_1) = \frac{1}{2} \cdot \frac{1}{2} \cdot r = \frac{1}{4} r$. Since $V^{(1)}(s_1) + V^{(2)}(s_1) = \frac{10}{16} r > \frac{8}{16} r = 2V^{\text{mid}}(s_1)$, the function $V^{\pi_\theta}$ is not concave in $\theta$. □

Non-concavity means gradient ascent can only guarantee convergence to stationary points in general. This motivates two directions: (a) studying structural conditions (e.g., softmax parameterization, regularization) under which stationary points are near-global optima, and (b) using the natural gradient to obtain faster, dimension-free convergence.

# Monte Carlo Estimation and REINFORCE

## Unbiased Gradient Estimation

Even if we know the MDP $M$, computing the gradient may be computationally intensive. We can obtain unbiased estimates from sampled trajectories $\tau \sim \mathbb{P}_\mu^{\pi_\theta}$. Given a trajectory $\tau$,

**Algorithm 1** REINFORCE (Williams, 1992)

**Require:** Initial parameters $\theta_0$, learning rates $\{\eta_t\}_{t\geq 0}$, number of iterations $T$

1: **for** $t = 0, 1, \ldots, T-1$ **do**
2: Sample trajectory $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \ldots) \sim \mathbb{P}_\mu^{\pi_{\theta_t}}$
3: Compute $\widehat{Q^{\pi_{\theta_t}}}(s_h, a_h) = \sum_{h'=h}^{\infty} \gamma^{h'-h} r_{h'}$ for each $h$
4: Compute $\widehat{\nabla V^{\pi_{\theta_t}}}(\mu) = \sum_{h=0}^{\infty} \gamma^h \widehat{Q^{\pi_{\theta_t}}}(s_h, a_h) \nabla \log \pi_{\theta_t}(a_h \mid s_h)$
5: Update $\theta_{t+1} = \theta_t + \eta_t \widehat{\nabla V^{\pi_{\theta_t}}}(\mu)$
6: **end for**

define:

$$\widehat{Q^{\pi_\theta}}(s_t, a_t) \coloneqq \sum_{t'=t}^{\infty} \gamma^{t'-t} r(s_{t'}, a_{t'}), \qquad \widehat{\nabla V^{\pi_\theta}}(\mu) \coloneqq \sum_{t=0}^{\infty} \gamma^t \widehat{Q^{\pi_\theta}}(s_t, a_t) \nabla \log \pi_\theta(a_t \mid s_t).$$

**Lemma 3** (Unbiased gradient estimate (Agarwal et al., 2021, Lemma 9.7))**.** *We have:*

$$\mathbb{E}_{\tau \sim \mathbb{P}_\mu^{\pi_\theta}} \left[ \widehat{\nabla V^{\pi_\theta}}(\mu) \right] = \nabla V^{\pi_\theta}(\mu).$$

*Proof.* Observe:

$$\begin{aligned}
\mathbb{E}_{\tau \sim \mathbb{P}_\mu^{\pi_\theta}} \left[ \widehat{\nabla V^{\pi_\theta}}(\mu) \right] &= \mathbb{E}_{\tau \sim \mathbb{P}_\mu^{\pi_\theta}} \left[ \sum_{t=0}^{\infty} \gamma^t \widehat{Q^{\pi_\theta}}(s_t, a_t) \nabla \log \pi_\theta(a_t \mid s_t) \right] \\
&\overset{(a)}{=} \mathbb{E}_{\tau \sim \mathbb{P}_\mu^{\pi_\theta}} \left[ \sum_{t=0}^{\infty} \gamma^t \mathbb{E}\left[ \widehat{Q^{\pi_\theta}}(s_t, a_t) \mid s_t, a_t \right] \nabla \log \pi_\theta(a_t \mid s_t) \right] \\
&\overset{(b)}{=} \mathbb{E}_{\tau \sim \mathbb{P}_\mu^{\pi_\theta}} \left[ \sum_{t=0}^{\infty} \gamma^t Q^{\pi_\theta}(s_t, a_t) \nabla \log \pi_\theta(a_t \mid s_t) \right] \\
&\overset{(c)}{=} \nabla V^{\pi_\theta}(\mu),
\end{aligned}$$

where $(a)$ follows from the tower property of conditional expectations, $(b)$ follows from the Markov property: $\mathbb{E}[\widehat{Q^{\pi_\theta}}(s_t, a_t) \mid s_t, a_t] = Q^{\pi_\theta}(s_t, a_t)$, and $(c)$ is the action-value form of the policy gradient theorem (Theorem 1(ii)). □

## The REINFORCE Algorithm

## Convergence to Stationary Points

We use the shorthand $\pi^{(t)} \coloneqq \pi_{\theta_t}$ and $V^{(t)} \coloneqq V^{\pi_{\theta_t}}$.

**Definition 3** ($\beta$-smoothness)**.** *A differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is $\beta$-smooth if*

$$\|\nabla f(w) - \nabla f(w')\| \le \beta \|w - w'\|$$

*for all $w, w' \in \mathbb{R}^d$. Intuitively, $\beta$-smoothness says the gradient cannot change too fast: nearby points have similar gradients, so the function has no sharp "kinks." A standard consequence is the* descent lemma*: for all $w, w'$,*

$$|f(w') - f(w) - \langle \nabla f(w),\, w' - w \rangle| \le \frac{\beta}{2} \|w' - w\|^2. \tag{1}$$

*In other words, the first-order Taylor approximation $f(w) + \langle \nabla f(w), w' - w \rangle$ remains a good approximation to $f(w')$ when $w'$ is close to $w$, with the approximation error growing at most quadratically in $\|w' - w\|$. This is what makes gradient-based optimization reliable: a gradient step of size $\eta$ changes the function value in a predictable way, with bounded error of order $\eta^2$.*

**Theorem 4** (Stochastic convergence to stationary points (Ghadimi and Lan, 2013))**.** *Assume that $V^{\pi_\theta}(\mu)$ is $\beta$-smooth in $\theta$. Let $M := V^*(\mu) - V^{(0)}(\mu)$. Suppose we run stochastic gradient ascent:*

$$\theta_{t+1} = \theta_t + \eta \widehat{\nabla V^{(t)}}(\mu),$$

*where the stochastic gradient satisfies:*

1. ***Unbiasedness:*** $\mathbb{E}\big[\widehat{\nabla V^{(t)}}(\mu) \mid \theta_t\big] = \nabla_\theta V^{(t)}(\mu)$*;*
2. ***Bounded second moment:*** $\mathbb{E}\big[\big\|\widehat{\nabla V^{(t)}}(\mu)\big\|^2 \mid \theta_t\big] \le \sigma^2$*.*

*Then with stepsize $\eta = \sqrt{2M/(\beta\sigma^2 T)}$:*

$$\min_{0 \le t < T} \mathbb{E}\left[\big\|\nabla_\theta V^{(t)}(\mu)\big\|^2\right] \le \sqrt{\frac{2\beta\sigma^2 M}{T}}.$$

*Proof.* **Step 1 (Per-step progress bound).** Applying the descent lemma (1) to $V^{\pi_\theta}(\mu)$ with $w = \theta_t$ and $w' = \theta_{t+1} = \theta_t + \eta \widehat{\nabla V^{(t)}}(\mu)$:

$$\left| V^{(t+1)}(\mu) - V^{(t)}(\mu) - \eta \big\langle \nabla_\theta V^{(t)}(\mu),\, \widehat{\nabla V^{(t)}}(\mu) \big\rangle \right| \le \frac{\beta\eta^2}{2} \big\|\widehat{\nabla V^{(t)}}(\mu)\big\|^2.$$

Removing the absolute value via the lower bound:

$$\eta \big\langle \nabla_\theta V^{(t)}(\mu),\, \widehat{\nabla V^{(t)}}(\mu) \big\rangle \le V^{(t+1)}(\mu) - V^{(t)}(\mu) + \frac{\beta\eta^2}{2} \big\|\widehat{\nabla V^{(t)}}(\mu)\big\|^2. \tag{2}$$

**Step 2 (Conditional expectation).** Taking $\mathbb{E}[\cdot \mid \theta_t]$ on both sides of (2). For the left-hand

side, since $\nabla_\theta V^{(t)}(\mu)$ is deterministic given $\theta_t$, unbiasedness gives:

$$\mathbb{E}\Big[\big\langle \nabla_\theta V^{(t)}(\mu),\, \widehat{\nabla V^{(t)}}(\mu)\big\rangle \,\Big|\, \theta_t\Big] = \big\langle \nabla_\theta V^{(t)}(\mu),\, \nabla_\theta V^{(t)}(\mu)\big\rangle = \big\|\nabla_\theta V^{(t)}(\mu)\big\|^2.$$

For the right-hand side, the bounded second moment assumption gives $\mathbb{E}\big[\|\widehat{\nabla V^{(t)}}(\mu)\|^2 \mid \theta_t\big] \le \sigma^2$. Combining:

$$\eta\big\|\nabla_\theta V^{(t)}(\mu)\big\|^2 \le \mathbb{E}\big[V^{(t+1)}(\mu) - V^{(t)}(\mu) \mid \theta_t\big] + \frac{\beta\eta^2\sigma^2}{2}. \tag{3}$$

**Step 3 (Telescope and sum).** Taking full expectation of (3) and summing over $t = 0, \dots, T-1$:

$$\eta\sum_{t=0}^{T-1}\mathbb{E}\big[\big\|\nabla_\theta V^{(t)}(\mu)\big\|^2\big] \le \underbrace{\sum_{t=0}^{T-1}\mathbb{E}\big[V^{(t+1)}(\mu) - V^{(t)}(\mu)\big]}_{=\mathbb{E}[V^{(T)}(\mu)]-V^{(0)}(\mu)\le M} + \frac{T\beta\eta^2\sigma^2}{2}.$$

Dividing both sides by $\eta T$:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\big[\big\|\nabla_\theta V^{(t)}(\mu)\big\|^2\big] \le \frac{M}{\eta T} + \frac{\beta\eta\sigma^2}{2}. \tag{4}$$

**Step 4 (Optimize stepsize).** The right-hand side of (4) has the form $A/\eta + B\eta$ with $A = M/T$ and $B = \beta\sigma^2/2$, which is minimized at $\eta^* = \sqrt{A/B} = \sqrt{2M/(\beta\sigma^2 T)}$, yielding:

$$\frac{M}{\eta^* T} + \frac{\beta\eta^*\sigma^2}{2} = 2\sqrt{AB} = 2\sqrt{\frac{M\beta\sigma^2}{2T}} = \sqrt{\frac{2\beta\sigma^2 M}{T}}.$$

The result follows since $\min_t \le \frac{1}{T}\sum_t$. □

**Remark 1.** *With exact gradients, using the constant stepsize $\eta = 1/\beta$ and substituting $\widehat{\nabla V^{(t)}}(\mu) = \nabla_\theta V^{(t)}(\mu)$ directly in Step 1 gives the faster deterministic rate $\min_{t<T}\|\nabla_\theta V^{(t)}(\mu)\|^2 \le 2\beta M/T$ (Agarwal et al., 2021, Lemma 9.6).*

**Remark 2.** *Theorem 4 guarantees convergence to a* stationary point *(i.e., a point where $\nabla_\theta V^{\pi_\theta}(\mu) = 0$). This is the best one can hope for in general non-concave optimization. For general non-concave functions, stationary points can be saddle points or local maxima that are far from the global optimum. The fact that the policy optimization landscape is non-concave means these results do not guarantee finding the optimal policy. Sections below on softmax global convergence and NPG overcome this limitation using structure specific to the RL problem.*

REINFORCE with exact gradients converges to a stationary point at rate $O(1/T)$. With stochastic gradients, the rate becomes $O(1/\sqrt{T})$ due to the second moment $\sigma^2$ of the gradient estimator. Both rates are standard for smooth non-convex optimization and cannot be improved in general (Ghadimi and Lan, 2013). The main bottleneck in practice is $\sigma^2$, which can be very large for long-horizon MDPs with sparse rewards — motivating the variance reduction techniques in the next section.

# Variance Reduction

## Baselines

The variance of REINFORCE can be large. To see why, recall that the gradient estimate involves the product $\widehat{Q}(s_t, a_t)\nabla \log \pi_\theta(a_t \mid s_t)$, where $\widehat{Q}$ is a sum of discounted future rewards. In a sparse-reward setting, $\widehat{Q}$ is often zero (no reward encountered), but occasionally very large (when the trajectory happens to reach a rewarding state). This creates high-variance gradient estimates. A common variance reduction technique is to subtract a state-dependent *baseline* $f(s)$.

**Lemma 5** (Baseline invariance)**.** *For any baseline function $f : \mathcal{S} \to \mathbb{R}$, if the samples used to construct $f$ are independent of the trajectory $\tau$ used to construct $\widehat{Q^{\pi_\theta}}(s_t, a_t)$, then:*

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \Big(\widehat{Q^{\pi_\theta}}(s_t, a_t) - f(s_t)\Big) \nabla \log \pi_\theta(a_t \mid s_t)\right] = \nabla V^{\pi_\theta}(\mu).$$

*Proof.* For any function $g(s)$:

$$\mathbb{E}_{a\sim\pi(\cdot|s)}[\nabla \log \pi(a \mid s)\, g(s)] = g(s) \sum_a \nabla \pi(a \mid s) = g(s)\nabla \sum_a \pi(a \mid s) = g(s)\nabla 1 = 0.$$

Since $f(\cdot)$ is independent of $\tau$, we have $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t f(s_t)\nabla \log \pi_\theta(a_t \mid s_t)\right] = 0$. The result follows from Lemma 3. □

The natural choice of baseline is $f(s) = V^\pi(s)$, so that $\widehat{Q^{\pi_\theta}}(s_t, a_t) - f(s_t)$ estimates the advantage $A^{\pi_\theta}(s_t, a_t)$. The advantage-form gradient has lower variance because $\mathbb{E}_a[A^\pi(s, a)] = 0$ for every state.

**Policy gradient with baseline.**

$$\nabla V^{\pi_\theta}(\mu) = \frac{1}{1-\gamma} \mathbb{E}_{s\sim d^{\pi_\theta}, a\sim\pi_\theta}\left[A^{\pi_\theta}(s, a)\nabla \log \pi_\theta(a \mid s)\right].$$

## Actor-Critic

In practice, the value function $V^\pi(s)$ used as the baseline must also be estimated. The *actor-critic* architecture maintains two components:

- **Actor**: the parameterized policy $\pi_\theta$, updated via policy gradient.
- **Critic**: a learned value function $\widehat{V}_\omega(s) \approx V^{\pi_\theta}(s)$ with parameters $\omega$, used to form the advantage estimate.

The critic is trained using the policy evaluation methods from Lecture 8 (e.g., TD learning, LSTD). This connects our work on policy evaluation to policy optimization.

The advantage is estimated using the *temporal difference (TD) residual*:

$$\widehat{A}(s_t, a_t) = r(s_t, a_t) + \gamma \widehat{V}_\omega(s_{t+1}) - \widehat{V}_\omega(s_t).$$

This is a one-step bootstrap estimate. More generally, one can use the *generalized advantage estimator* (GAE), which interpolates between the high-bias one-step TD estimate and the high-variance Monte Carlo estimate via a parameter $\lambda \in [0, 1]$:

$$\widehat{A}^{\mathrm{GAE}(\lambda)}(s_t, a_t) = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}, \quad \text{where} \quad \delta_t = r(s_t, a_t) + \gamma \widehat{V}_\omega(s_{t+1}) - \widehat{V}_\omega(s_t).$$

When $\lambda = 0$, this reduces to the one-step TD residual. When $\lambda = 1$, it recovers the Monte Carlo advantage estimate (with the baseline $\widehat{V}_\omega$). In practice, $\lambda \in [0.9, 0.99]$ gives a good bias-variance trade-off.

The actor-critic gradient estimate is:

$$\widehat{\nabla V}(\mu) = \sum_{t=0}^{T-1} \gamma^t \widehat{A}^{\mathrm{GAE}(\lambda)}(s_t, a_t) \nabla \log \pi_\theta(a_t \mid s_t).$$

Unlike the REINFORCE estimate (which uses $\widehat{Q}$ from full trajectory returns), this estimate is *biased* when $\widehat{V}_\omega \neq V^{\pi_\theta}$, but it typically has much lower variance because it does not depend on the full trajectory return.

Actor-critic = policy gradient (actor) + policy evaluation (critic, Lecture 8). The critic provides low-variance advantage estimates at the cost of introducing bias from function approximation. The bias-variance trade-off is controlled by $\lambda$: $\lambda \to 0$ reduces variance but increases bias; $\lambda \to 1$ reduces bias but increases variance.

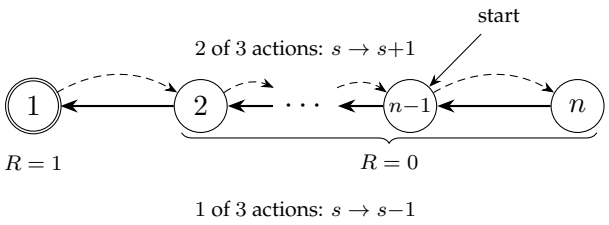
# Conservative Policy Iteration

A fundamental challenge in policy optimization is the *state distribution shift*: when we change the policy, the states we visit change as well. This coupling between the policy and its induced state distribution makes policy optimization fundamentally different from standard optimization. Before introducing the CPI algorithm, we examine two examples that reveal concrete failure modes of the policy gradient approach.

## Motivating Examples

The following examples from Kakade and Langford (2002) illustrate two failure modes of policy gradient methods arising from the dependence on the current state distribution $d^\pi$.

**Example 1: Exponential sample complexity in chain MDPs.**



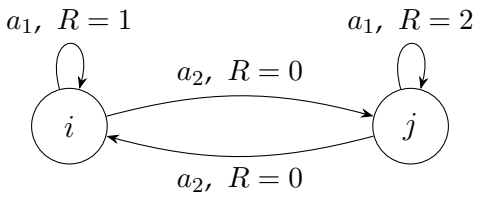

Consider a chain MDP with states $\{1, 2, \ldots, n\}$ and 3 actions per state. Two actions deterministically move to $s + 1$ (away from the goal), while one action moves to $s - 1$ (toward the goal). Rewards are 0 everywhere except state 1, where $R = 1$. The optimal policy always takes the backward action.

Starting from state $n - 1$ with the uniform policy $\pi(a \mid s) = 1/3$, the agent takes two steps away from the goal for every step toward it on average—a biased random walk. By the Gambler's ruin, the probability of reaching any state $s \le n/2$ starting from state $n - 1$ is $\left(\frac{1/3}{2/3}\right)^{n/2} = (1/2)^{n/2}$, exponentially small in $n$. This has two consequences for the policy gradient $\nabla V^\pi(\mu) = \frac{1}{1-\gamma} \sum_s d^\pi(s) \sum_a Q^\pi(s, a) \nabla \pi(a \mid s)$:

- For states $s \le n/2$ (near the goal): $d^\pi(s) \le (1/2)^{n/2}$, so the *visitation* of these states is exponentially small.
- For states $s \ge n/2$ (far from the goal): since rewards are zero except at state 1, we have $Q^\pi(s, a) \le \mathbb{P}^\pi(\text{reaching state } 1 \mid s)$. By the same Gambler's ruin argument, this probability is at most $(1/2)^{n/2}$ from any state $s \ge n/2$.

Every component of the gradient has magnitude at most $\frac{n}{1-\gamma}(1/2)^{n/2}$. Obtaining meaningful gradient estimates requires exponentially many sample trajectories.

**Example 2: Policy gradient can worsen the policy.**




Consider a 2-state MDP with states $\{i, j\}$ and actions $\{a_1, a_2\}$. Action $a_1$ is a self-loop with reward $1$ in state $i$ and reward $2$ in state $j$; action $a_2$ transitions to the other state with reward $0$. The optimal policy plays $a_2$ in state $i$ (transition to $j$) and $a_1$ in state $j$ (stay and collect reward $2$).

Define the initial policy:

$$\pi(a_1 \mid i) = 0.8,\ \pi(a_2 \mid i) = 0.2, \qquad \pi(a_1 \mid j) = 0.2,\ \pi(a_2 \mid j) = 0.8.$$

Its stationary distribution is $p(i) = 0.8$, $p(j) = 0.2$.

**Computing Q-values and advantages.** The Bellman equations are:

$$V^\pi(i) = 0.8\big(1 + \gamma V^\pi(i)\big) + 0.2\,\gamma V^\pi(j),$$
$$V^\pi(j) = 0.2\big(2 + \gamma V^\pi(j)\big) + 0.8\,\gamma V^\pi(i).$$

Subtracting the two equations gives $V^\pi(i) - V^\pi(j) = 0.4$. The Q-values at state $i$ are:

$$Q^\pi(i, a_1) = 1 + \gamma V^\pi(i), \qquad Q^\pi(i, a_2) = \gamma V^\pi(j),$$

so $Q^\pi(i, a_1) - Q^\pi(i, a_2) = 1 + 0.4\gamma > 0$, i.e., $A^\pi(i, a_1) > 0$.

**The gradient update goes in the wrong direction.** With softmax parameterization $\pi_\theta(a \mid s) \propto e^{\theta_{s,a}}$, the policy gradient theorem gives $\frac{\partial V}{\partial \theta_{s,a}} = \frac{1}{1-\gamma} d^\pi(s)\pi(a \mid s)A^\pi(s, a)$ for softmax policies (this follows from the advantage form of the policy gradient and the softmax score function), so the gradient update is $\theta_{s,a} \leftarrow \theta_{s,a} + \eta\, d^\pi(s)\, \pi(a \mid s)\, A^\pi(s, a)$. Since $A^\pi(i, a_1) > 0$, the update *increases* $\theta_{i,a_1}$, making the policy loop more on state $i$—the opposite of the optimal action $a_2$. Meanwhile, the correct update at state $j$ (where $A^\pi(j, a_1) > 0$) has much smaller magnitude because $d^\pi(i) \gg d^\pi(j)$.

This creates a vicious cycle: the updated policy visits $j$ even less $\to$ $d^\pi(j)$ shrinks $\to$ the useful gradient signal from $j$ vanishes $\to$ the wrong update at $i$ dominates further. Figure 2 shows $d^\pi(j)$ dropping by nearly 7 orders of magnitude before recovering after $\sim 2 \times 10^7$ iterations.

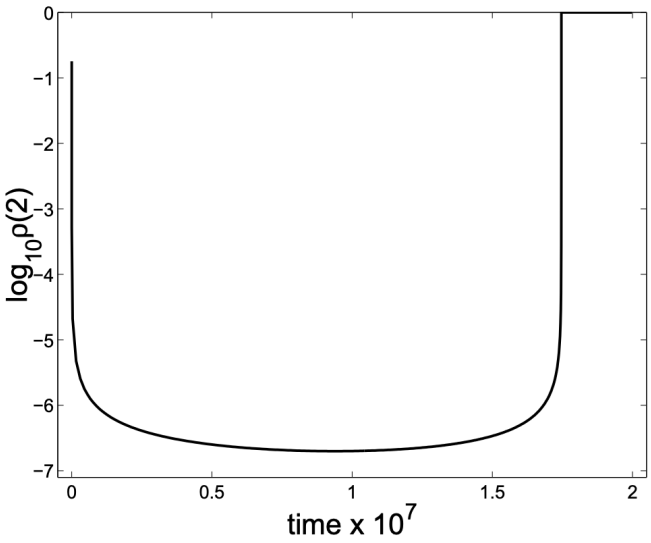



Figure 2: Stationary probability of state $j$ under policy gradient updates (Kakade and Langford, 2002).

Both examples reveal the same root cause: policy gradient updates are weighted by the *current* state distribution $d^\pi$. If the current policy poorly explores the states that matter for improvement, the gradient signal is either vanishingly small (Example 1) or misleading (Example 2). CPI addresses this by making small, controlled updates that guarantee monotonic improvement regardless of the distribution shift.

We recall the performance difference lemma from a previous lecture, which is central to the analysis below.

**Lemma 6** (Performance difference lemma (Kakade and Langford, 2002))**.** *For any two policies $\pi$ and $\widetilde{\pi}$,*

$$V^{\widetilde{\pi}}(\mu) - V^{\pi}(\mu) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\mu^{\widetilde{\pi}}}\sum_a \widetilde{\pi}(a \mid s)\, A^{\pi}(s,a).$$

## The CPI Algorithm

Conservative Policy Iteration (CPI) (Kakade and Langford, 2002) addresses the challenges above by controlling the state distribution shift through *mixture* updates instead of gradient updates. Both vanilla PG and NPG handle distribution shift implicitly; CPI takes a more explicit approach.

The key idea is simple. Instead of replacing $\pi^t$ with an entirely new policy, CPI mixes the

**Algorithm 2** Conservative Policy Iteration (CPI) (Kakade and Langford, 2002)

**Require:** Initial policy $\pi^0 \in \Pi$, accuracy parameter $\varepsilon$
1: **for** $t = 0, 1, 2, \ldots$ **do**
2: Compute greedy policy $\pi' = \mathcal{G}_\varepsilon(\pi^t, \Pi, \mu)$ ▷ *$\varepsilon$-approximate greedy policy selector*
3: **if** $\mathbb{E}_{s\sim d_\mu^{\pi^t}} A^{\pi^t}(s, \pi'(s)) \le \varepsilon$ **then**
4: **return** $\pi^t$
5: **end if**
6: Update $\pi^{t+1} = (1-\alpha)\pi^t + \alpha\pi'$ ▷ *mixture update with step size $\alpha$*
7: **end for**

current policy with a greedy improvement:

$$\pi^{t+1} = (1-\alpha)\pi^t + \alpha\pi',$$

where $\pi'$ is an (approximate) greedy policy with respect to $A^{\pi^t}$ and $\alpha \in (0,1)$ is a small step size. Because $\pi^{t+1}$ is close to $\pi^t$ in total variation ($\|\pi^{t+1} - \pi^t\|_\infty \le 2\alpha$), the state distributions $d^{\pi^{t+1}}$ and $d^{\pi^t}$ are also close, enabling us to bound the approximation error of the performance difference lemma (Lemma 6).

A notable advantage of CPI over gradient-based methods is that it works for *any* policy class $\Pi$, including non-differentiable or discrete policy classes, as long as one can compute (approximately) greedy policies.

Here $\mathcal{G}_\varepsilon(\pi, \Pi, \mu)$ denotes an *$\varepsilon$-approximate greedy policy selector*: it returns a policy $\pi' \in \Pi$ satisfying

$$\mathbb{E}_{s\sim d_\mu^\pi} \sum_a \pi'(a \mid s)\, A^\pi(s,a) \ge \max_{\widetilde{\pi}\in\Pi} \mathbb{E}_{s\sim d_\mu^\pi} \sum_a \widetilde{\pi}(a \mid s)\, A^\pi(s,a) - \varepsilon.$$

## Monotonic Improvement

**Lemma 7** (Similar policies ⇒ similar state visitations)**.** *Consider any $t$. We have:*

$$\|\pi^{t+1}(\cdot \mid s) - \pi^t(\cdot \mid s)\|_1 \le 2\alpha, \quad \forall s;$$

*and:*

$$\|d_\mu^{\pi^{t+1}} - d_\mu^{\pi^t}\|_1 \le \frac{2\alpha\gamma}{1-\gamma}.$$

*Proof.* The first claim follows from the definition of the mixture update:

$$\|\pi^{t+1}(\cdot \mid s) - \pi^t(\cdot \mid s)\|_1 = \alpha\|\pi'(\cdot \mid s) - \pi^t(\cdot \mid s)\|_1 \le 2\alpha.$$

For the second claim, let $\mathbb{P}_h^\pi$ denote the state distribution at time step $h$ under $\pi$ starting from $\mu$. We bound $\|\mathbb{P}_h^{\pi^{t+1}} - \mathbb{P}_h^{\pi^t}\|_1$ by induction. For $h \geq 1$:

$$\begin{aligned}\sum_{s'} |\mathbb{P}_h^{\pi^{t+1}}(s') - \mathbb{P}_h^{\pi^t}(s')| &= \sum_{s'} \left| \sum_{s,a} \left( \mathbb{P}_{h-1}^{\pi^{t+1}}(s)\pi^{t+1}(a \mid s) - \mathbb{P}_{h-1}^{\pi^t}(s)\pi^t(a \mid s) \right) P(s' \mid s,a) \right| \\ &\leq \sum_s \mathbb{P}_{h-1}^{\pi^{t+1}}(s) \sum_a |\pi^{t+1}(a \mid s) - \pi^t(a \mid s)| + \sum_s |\mathbb{P}_{h-1}^{\pi^{t+1}}(s) - \mathbb{P}_{h-1}^{\pi^t}(s)| \\ &\leq 2\alpha + \|\mathbb{P}_{h-1}^{\pi^{t+1}} - \mathbb{P}_{h-1}^{\pi^t}\|_1 \leq 2h\alpha.\end{aligned}$$

Using $d_\mu^\pi = (1-\gamma)\sum_{h=0}^\infty \gamma^h \mathbb{P}_h^\pi$:

$$\|d_\mu^{\pi^{t+1}} - d_\mu^{\pi^t}\|_1 \leq (1-\gamma)\sum_{h=0}^\infty \gamma^h \cdot 2h\alpha = 2\alpha(1-\gamma)\cdot\frac{\gamma}{(1-\gamma)^2} = \frac{2\alpha\gamma}{1-\gamma}.$$

□

**Theorem 8** (Monotonic improvement in CPI). *For any policy $\pi'$, let $\pi^{t+1} = (1-\alpha)\pi^t + \alpha\pi'$ and denote $\mathbb{A} = \mathbb{E}_{s\sim d_\mu^{\pi^t}} A^{\pi^t}(s, \pi'(s))$. We have:*

$$V_\mu^{\pi^{t+1}} - V_\mu^{\pi^t} \geq \frac{\alpha}{1-\gamma}\left(\mathbb{A} - \frac{2\alpha\gamma}{(1-\gamma)^2}\right).$$

*Setting $\alpha = \frac{\mathbb{A}(1-\gamma)^2}{4\gamma}$:*

$$V_\mu^{\pi^{t+1}} - V_\mu^{\pi^t} \geq \frac{\mathbb{A}^2(1-\gamma)}{8\gamma}.$$

*Proof.* By the performance difference lemma (Lemma 6):

$$V_\mu^{\pi^{t+1}} - V_\mu^{\pi^t} = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\mu^{\pi^{t+1}}} \alpha A^{\pi^t}(s, \pi'(s)),$$

since $\pi^{t+1} = (1-\alpha)\pi^t + \alpha\pi'$ and $A^{\pi^t}(s, \pi^t(s)) = 0$. We have:

$$\begin{aligned}(1-\gamma)\left(V_\mu^{\pi^{t+1}} - V_\mu^{\pi^t}\right) &= \mathbb{E}_{s\sim d_\mu^{\pi^t}} \alpha A^{\pi^t}(s, \pi'(s)) + \mathbb{E}_{s\sim d_\mu^{\pi^{t+1}}} \alpha A^{\pi^t}(s, \pi'(s)) - \mathbb{E}_{s\sim d_\mu^{\pi^t}} \alpha A^{\pi^t}(s, \pi'(s)) \\ &\geq \alpha\mathbb{A} - \alpha \sup_{s,a} |A^{\pi^t}(s,a)| \cdot \|d_\mu^{\pi^{t+1}} - d_\mu^{\pi^t}\|_1 \\ &\geq \alpha\mathbb{A} - \frac{\alpha}{1-\gamma}\cdot\frac{2\alpha\gamma}{1-\gamma} = \alpha\left(\mathbb{A} - \frac{2\alpha\gamma}{(1-\gamma)^2}\right),\end{aligned}$$

where the second inequality uses $|A^{\pi^t}(s,a)| \leq 1/(1-\gamma)$ and Lemma 7. Optimizing over $\alpha$ gives $\alpha = \frac{\mathbb{A}(1-\gamma)^2}{4\gamma}$, yielding the second claim. □

Note that the above bound holds for *any* choice of $\pi'$—it does not depend on the approximation quality $\varepsilon$ of $\mathcal{G}_\varepsilon$. The role of $\varepsilon$ only appears when we analyze termination: at that point, we need $\mathcal{G}_\varepsilon$ to certify that *no* policy in $\Pi$ has large advantage, not just the particular $\pi'$ returned.

## Convergence and Global Optimality

**Theorem 9** (Local optimality). *Algorithm 2 terminates in at most $8\gamma/\varepsilon^2$ steps and outputs a policy $\pi^t$ satisfying $\max_{\pi\in\Pi} \mathbb{E}_{s\sim d_\mu^{\pi^t}} A^{\pi^t}(s,\pi(s)) \le 2\varepsilon$.*

*Proof.* The reward is bounded in $[0,1]$, so $V_\mu^\pi \in [0, 1/(1-\gamma)]$. By Theorem 8, every iteration $t$ where the algorithm does not terminate (i.e., $\mathbb{A} > \varepsilon$) achieves improvement at least $\varepsilon^2(1-\gamma)/(8\gamma)$. Since total improvement is at most $1/(1-\gamma)$, the algorithm terminates in at most $\frac{1/(1-\gamma)}{\varepsilon^2(1-\gamma)/(8\gamma)} = 8\gamma/\varepsilon^2$ steps. At termination, the greedy policy $\pi' = \mathcal{G}_\varepsilon(\pi^t, \Pi, \mu)$ satisfies $\max_\pi \mathbb{E}_{s\sim d_\mu^{\pi^t}} A^{\pi^t}(s,\pi(s)) \le \mathbb{A} + \varepsilon \le 2\varepsilon$. □

**Theorem 10** (Global optimality). *Upon termination, we have a policy $\pi$ such that:*

$$V^\star(\mu) - V^\pi(\mu) \le \frac{2\varepsilon + \epsilon_\Pi}{(1-\gamma)^2} \left\| \frac{d_\mu^{\pi^\star}}{\mu} \right\|_\infty,$$

*where $\epsilon_\Pi := \mathbb{E}_{s\sim d_\mu^\pi}[\max_{a\in\mathcal{A}} A^\pi(s,a)] - \max_{\widetilde{\pi}\in\Pi} \mathbb{E}_{s\sim d_\mu^\pi}[A^\pi(s,\widetilde{\pi}(s))]$ measures the expressiveness gap of $\Pi$.*

*Proof.* By the performance difference lemma (Lemma 6):

$$\begin{aligned} V^\star(\mu) - V^\pi(\mu) &\le \frac{1}{1-\gamma} \mathbb{E}_{s\sim d_\mu^{\pi^\star}} \max_{a\in\mathcal{A}} A^\pi(s,a) \\ &\le \frac{1}{1-\gamma} \left\| \frac{d_\mu^{\pi^\star}}{d_\mu^\pi} \right\|_\infty \mathbb{E}_{s\sim d_\mu^\pi} \max_{a\in\mathcal{A}} A^\pi(s,a) \\ &\le \frac{1}{(1-\gamma)^2} \left\| \frac{d_\mu^{\pi^\star}}{\mu} \right\|_\infty \mathbb{E}_{s\sim d_\mu^\pi} \max_{a\in\mathcal{A}} A^\pi(s,a), \end{aligned}$$

where the last step uses $d_\mu^\pi(s) \ge (1-\gamma)\mu(s)$. By the definition of $\epsilon_\Pi$ and Theorem 9:

$$\mathbb{E}_{s\sim d_\mu^\pi} \max_a A^\pi(s,a) = \max_{\widetilde{\pi}\in\Pi} \mathbb{E}_{s\sim d_\mu^\pi}[A^\pi(s,\widetilde{\pi}(s))] + \epsilon_\Pi \le 2\varepsilon + \epsilon_\Pi.$$

□

CPI vs. PG: CPI uses mixture updates (works for any policy class, not just differentiable ones), while PG uses gradient updates. Both achieve local optimality; both need the distribution mismatch coefficient $\|d_\mu^{\pi^\star}/\mu\|_\infty$ for global optimality guarantees.

**Remark 3** (The distribution mismatch coefficient)**.** *The factor $\|d_\mu^{\pi^\star}/\mu\|_\infty$ appears in the CPI bound (Theorem 10) and more broadly in global convergence bounds for policy gradient methods. It measures how well the initial distribution $\mu$ covers the states that the optimal policy $\pi^\star$ visits. If $\mu$ is concentrated on states that $\pi^\star$ rarely visits, then $\|d_\mu^{\pi^\star}/\mu\|_\infty$ is large, and the guarantees degrade. This is a fundamental quantity in RL that cannot be avoided without additional assumptions (such as access to a simulator or a reset distribution). It reflects the inherent difficulty of learning from a fixed starting distribution: if the learner's data comes from the wrong part of the state space, no amount of optimization can compensate.*

*Notably, the natural policy gradient (NPG) convergence bound (Kakade, 2001) does* not *depend on this coefficient in the tabular softmax setting, because the Fisher information cancels the state distribution from the update. However, once we move to function approximation, the distribution mismatch reappears.*

# Trust Region Policy Optimization and PPO

Like CPI, Trust Region Policy Optimization (TRPO) and Proximal Policy Optimization (PPO) enforce conservative policy updates to control the state distribution shift. While CPI uses mixture updates, TRPO and PPO constrain the KL divergence or probability ratio between successive policies.

## TRPO

TRPO (Schulman et al., 2015) starts from the performance difference lemma (Lemma 6), which tells us $V^{\pi'}(\mu) - V^{\pi}(\mu) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\mu^{\pi'}} \sum_a \pi'(a \mid s) A^\pi(s,a)$. The difficulty is that this expression depends on $d_\mu^{\pi'}$, the state distribution of the *new* policy, which is unknown before we commit to $\pi'$. TRPO replaces $d_\mu^{\pi'}$ with the *current* distribution $d_\mu^{\pi_{\theta_t}}$, defining the surrogate objective:

$$L_{\theta_t}(\theta) := V^{\pi_{\theta_t}}(\mu) + \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\mu^{\pi_{\theta_t}}} \mathbb{E}_{a\sim\pi_\theta(\cdot|s)} A^{\pi_{\theta_t}}(s,a).$$

Since the surrogate replaces the state distribution $d_\mu^{\pi_\theta}$ of the new policy with $d_\mu^{\pi_{\theta_t}}$ of the old policy, $L_{\theta_t}(\theta) \neq V^{\pi_\theta}(\mu)$ in general. The two agree at $\theta = \theta_t$ (where the distributions coincide), and in fact $L_{\theta_t}$ matches $V^{\pi_\theta}(\mu)$ to first order (i.e., $L_{\theta_t}(\theta_t) = V^{\pi_{\theta_t}}(\mu)$ and $\nabla L_{\theta_t}(\theta_t) = \nabla V^{\pi_{\theta_t}}(\mu)$), but $L_{\theta_t}$ can be a poor approximation far from $\theta_t$. The advantage is that $L_{\theta_t}$ can be estimated

entirely from data collected under $\pi_{\theta_t}$; the price is that we must stay close to $\theta_t$. The CPI analysis (cf. Theorem 8) quantifies this: $V^{\pi_\theta}(\mu) \geq L_{\theta_t}(\theta) - C \cdot \max_s \mathrm{KL}(\pi_{\theta_t}(\cdot \mid s) \| \pi_\theta(\cdot \mid s))$ with $C = \frac{4\gamma}{(1-\gamma)^2} \max_{s,a} |A^{\pi_{\theta_t}}(s,a)|$. One could maximize the right-hand side directly (a minorization-maximization scheme), but $C$ is very conservative in practice, leading to tiny steps. TRPO replaces the penalty with a hard constraint on the average KL divergence:

$$\begin{aligned} \max_{\theta} \quad & L_{\theta_t}(\theta) \\ \text{s.t.} \quad & \mathbb{E}_{s \sim d_\mu^{\pi_{\theta_t}}} \mathrm{KL}\big(\pi_{\theta_t}(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid s)\big) \leq \delta. \end{aligned}$$

The trust region radius $\delta$ is easier to tune than the penalty coefficient $C$, and empirically allows much larger steps while maintaining monotonic improvement. Approximating this constrained problem via Taylor expansion introduces the *Fisher information matrix* and yields the *natural policy gradient* direction $F_{\theta_t}^{-1} \nabla V^{\pi_{\theta_t}}$; we derive this connection in a future lecture.

## PPO

Proximal Policy Optimization (PPO) (Schulman et al., 2017) was developed as a simpler alternative to TRPO. While TRPO requires conjugate gradient solvers and Fisher-vector products to approximately solve the KL-constrained problem, PPO achieves a similar effect using only first-order optimization.

**Importance-weighted surrogate.** Using importance sampling to replace $a \sim \pi_\theta$ with $a \sim \pi_{\theta_t}$, the surrogate objective can be written as an expectation under the current policy (dropping the constant $V^{\pi_{\theta_t}}(\mu)$ and the $\frac{1}{1-\gamma}$ scaling, which do not affect optimization):

$$\widehat{L}_{\theta_t}(\theta) \coloneqq \mathbb{E}_{s \sim d_\mu^{\pi_{\theta_t}}, a \sim \pi_{\theta_t}} \left[ r_t(\theta)\, A^{\pi_{\theta_t}}(s,a) \right], \qquad r_t(\theta) \coloneqq \frac{\pi_\theta(a \mid s)}{\pi_{\theta_t}(a \mid s)}.$$

Maximizing $\widehat{L}_{\theta_t}$ without any constraint would lead to excessively large policy updates (large $r_t$), making the surrogate a poor approximation. One could add a KL penalty $\widehat{L}_{\theta_t}(\theta) - \beta\, \mathrm{KL}(\pi_{\theta_t} \| \pi_\theta)$, but in practice it is hard to choose a fixed $\beta$ that works well throughout training (Schulman et al., 2017).

**Clipped surrogate.** PPO instead enforces a soft trust region by clipping the probability ratio:

$$L^{\mathrm{CLIP}}(\theta) = \mathbb{E}_{s \sim d_\mu^{\pi_{\theta_t}}, a \sim \pi_{\theta_t}} \min \left\{ r_t(\theta) \widehat{A}_t,\ \mathrm{clip}(r_t(\theta),\, 1-\epsilon,\, 1+\epsilon)\, \widehat{A}_t \right\},$$

where $\mathrm{clip}(x, 1-\epsilon, 1+\epsilon) = \max(\min(x, 1+\epsilon), 1-\epsilon)$ and $\epsilon \approx 0.2$ is typical. The key design principle is that $L^{\mathrm{CLIP}}$ is a *pessimistic lower bound* on the unclipped surrogate $\widehat{L}_{\theta_t}$: the $\min$ removes any benefit from moving $r_t$ outside $[1-\epsilon,\, 1+\epsilon]$, but retains the full penalty when

the change hurts the objective. Concretely:

- When $\widehat{A}_t > 0$ (good action): the objective is $\min\{r_t, 1+\epsilon\} \cdot \widehat{A}_t$. The clipping removes the incentive to increase $r_t$ beyond $1+\epsilon$.
- When $\widehat{A}_t < 0$ (bad action): the objective is $\max\{r_t, 1-\epsilon\} \cdot \widehat{A}_t$. The clipping removes the incentive to decrease $r_t$ below $1-\epsilon$.

**Algorithm.** In each iteration, PPO:

(1) Collects a batch of trajectories using the current policy $\pi_{\theta_t}$.

(2) Estimates advantages $\widehat{A}_t$ (typically via generalized advantage estimation with a learned value function baseline).

(3) Performs $K$ epochs of mini-batch SGD on $L^{\mathrm{CLIP}}(\theta)$, reusing the same batch of data. The clipping prevents the policy from drifting too far despite multiple gradient steps on stale data—this is PPO's main practical advantage over vanilla policy gradient, which can only use each batch once.

PPO's simplicity and strong empirical performance have made it one of the most widely used policy optimization algorithms in practice, including in RLHF for large language models.

CPI, TRPO, and PPO all enforce conservative policy updates to prevent large state distribution shifts. CPI uses mixture updates (works for any policy class), TRPO uses KL-constrained updates with second-order Fisher information, and PPO uses clipping with first-order optimization. All three instantiate the principle: control the policy change to ensure monotonic improvement.

**Summary.** This lecture introduced policy optimization as a direct approach to finding good policies.

- The policy gradient theorem provides an expression for $\nabla V^{\pi_\theta}$ that can be estimated from trajectory samples.
- REINFORCE uses Monte Carlo returns for unbiased gradient estimation; actor-critic methods reduce variance via learned baselines (connecting to policy evaluation from Lecture 8).
- Non-concavity of $V^{\pi_\theta}$ in $\theta$ means gradient ascent can only guarantee convergence to stationary points in general.
- CPI, TRPO, and PPO enforce conservative updates to control state distribution shift, each with different mechanisms: mixture updates, KL-constrained second-order updates, and clipped first-order updates.

A natural next step is the *natural policy gradient* (Kakade, 2001), which replaces the Euclidean geometry in parameter space with the Fisher information geometry of policy space, achieving dimension-free global convergence and revealing a deep connection to TRPO.

# References

Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. *Reinforcement Learning: Theory and Algorithms*. Self-published, 2021. Available at https://rltheorybook.github.io/.

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Sham Kakade. A natural policy gradient. In *Advances in Neural Information Processing Systems*, volume 14, 2001.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, pages 267–274, 2002.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12, 1999.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992.