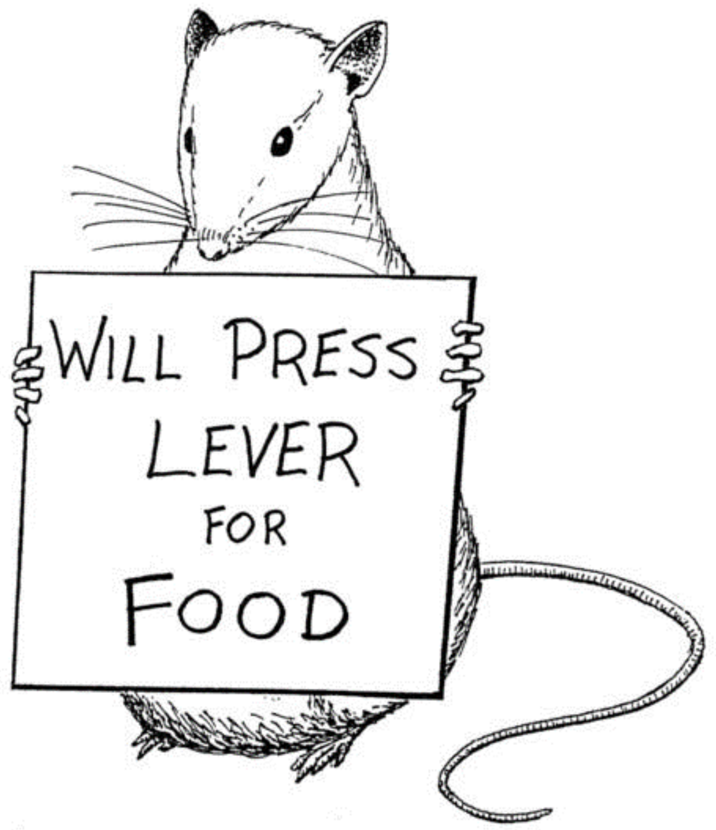
UCI University of California, Irvine

CS 277: Control and Reinforcement Learning

Winter 2024

Lecture 5: Policy-Gradient Methods

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



Logistics

assignments

- Quiz 1 has been graded
- Quiz 3 to be published soon, due next Monday
- We'll discuss Exercise 1 after the grace days
- Exercise 2 due the following Monday

Recap: policy evaluation

	model-based	model-free
Monte Carlo (MC)	$V_\pi(s_0) = \mathbb{E}_{\xi \sim p_\pi}[R \mid s_0]$	$\xi \sim p_\pi \qquad V(s_0) \to R(\xi)$
Temporal Difference (TD) (on-policy)	$V_\pi(s) = \mathbb{E}_{a \mid s \sim \pi}[r(s,a) + \gamma \mathbb{E}_{s' \mid s,a \sim p}[V_\pi(s')]]$	$s, a, r, s' \sim p_\pi \qquad V(s) \to r + \gamma V(s')$
Temporal Difference (TD) (off-policy)	$Q_\pi(s,a) = r(s,a) + \gamma \mathbb{E}_{\substack{s' \mid s,a \sim p \\ a' \mid s' \sim \pi}}[Q_\pi(s',a')]]$	$s, a, r, s' \sim p_{\pi'} \quad Q(s,a) \to r + \gamma \mathbb{E}_{a' \mid s' \sim \pi}[Q(s',a')]$

Recap: policy ~~evaluation~~ improvement

	model-based	model-free
Monte Carlo (MC)	$V_\pi(s_0) = \mathbb{E}_{\xi\sim p_\pi}[R \mid s_0]$	$\xi \sim p_\pi$ $\quad V(s_0) \to R(\xi)$
Temporal Difference (TD) (on-policy)	$V_\pi(s) = \overset{\max_a}{\cancel{\mathbb{E}_{a\mid s\sim\pi}}}[r(s,a) + \gamma\mathbb{E}_{s'\mid s,a\sim p}[V_\pi(s')]]$ **Value Iteration**	$s,a,r,s' \sim p_\pi$ $\quad V(s) \to r + \gamma V(s')$
Temporal Difference (TD) (off-policy)	$Q_\pi(s,a) = r(s,a) + \gamma\mathbb{E}_{\substack{s'\mid s,a\,\sim\,p \\ \cancel{a'\mid s'\sim\pi} \\ \max_{a'}}}[Q_\pi(s',a')]]$	$s,a,r,s' \sim p_{\pi'}$ $\quad Q(s,a) \to r + \gamma\overset{\max_{a'}}{\cancel{\mathbb{E}_{a'\mid s'\sim\pi}}}[Q(s',a')]$ **Q-learning**

Deep Q-Learning (DQN)

```
**Algorithm** DQN
Initialize $\theta$, 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 $T_{\text{target}}$ steps, set $\bar{\theta} \leftarrow \theta$
    $s \leftarrow$ reset state if $s'$ terminal, else $s \leftarrow s'$
```

MF

θ

DP

π'

max

[Mnih et al., 2015]

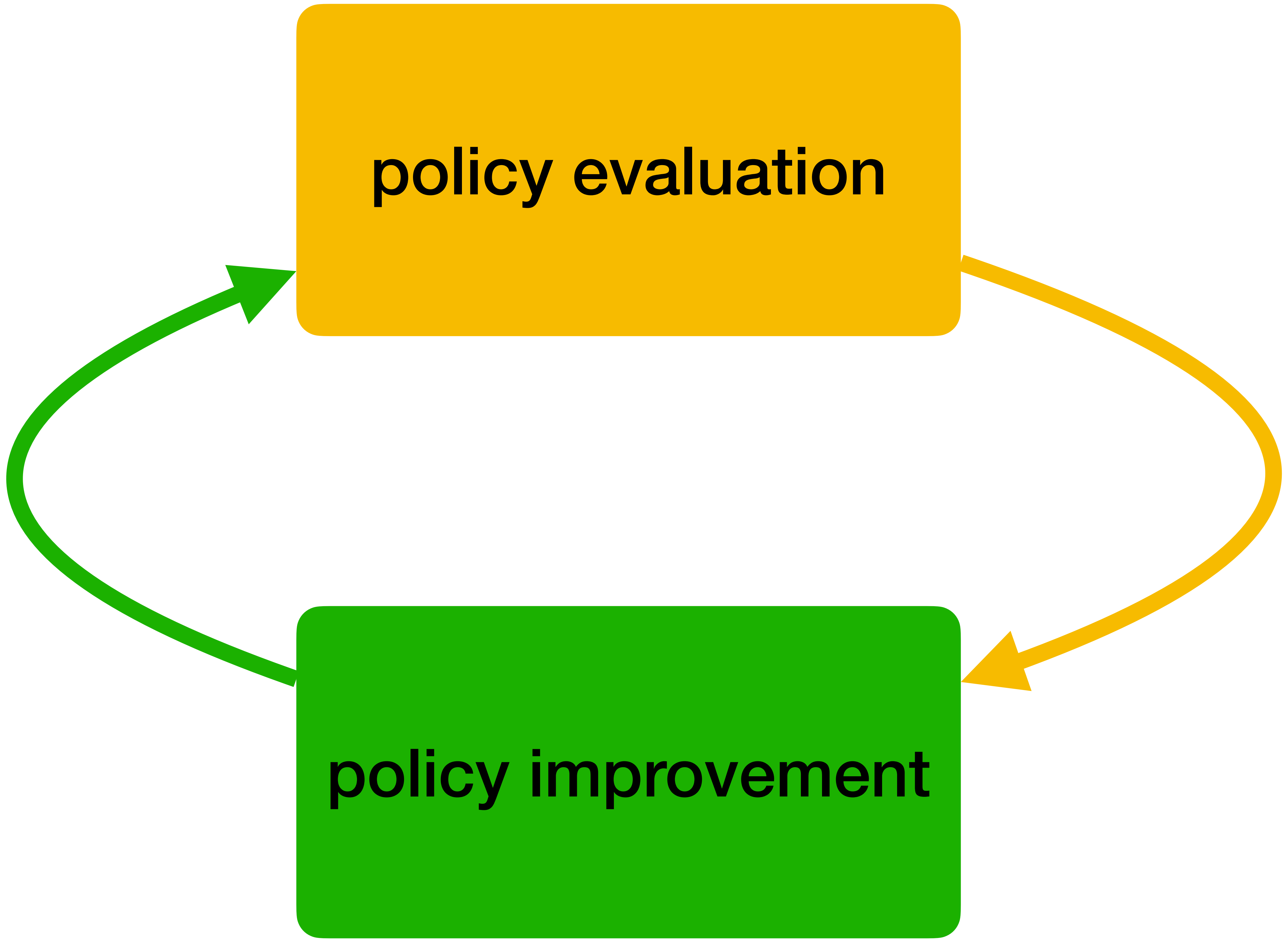
Today's lecture

Policy Gradient

Actor–Critic PG

Advantage estimation

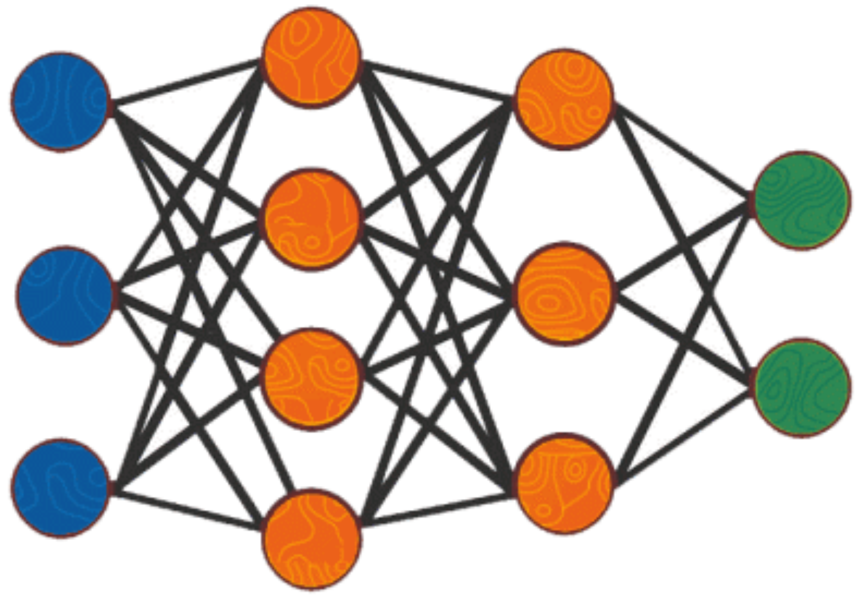
Value-based vs. policy-based methods



value-based		policy-based
$Q_\theta(s, a)$	policy evaluation	$\mathbb{E}_{\xi \sim p_\theta}[R(\xi)]$
$\arg\max_a Q_\theta(s, a)$	policy improvement	$\pi_\theta(a \mid s)$

Policy Gradient (PG)

- Gradient-based learning: $\theta \to \theta - \nabla_\theta \mathbb{E}_{x \sim D}[\mathcal{L}_\theta(x)]$
  - Can estimate expectation with samples



- Policy-Gradient RL: $\theta \to \theta + \nabla_\theta J_\theta$, with $J_\theta = \mathbb{E}_{\xi \sim p_\theta}[R]$
  - Can we also use samples $\xi \sim p_\theta$?
- The sampling distribution itself depends on θ
  - Data must be on-policy
  - Cannot backprop gradient through samples

Score-function gradient estimation

- Log-derivative + chain rule: $\nabla_\theta \log p_\theta(\xi) = \dfrac{1}{p_\theta(\xi)} \nabla_\theta p_\theta(\xi)$

- Log-derivative / score-function / REINFORCE trick:

$$\begin{aligned}
\nabla_\theta J_\theta &= \sum_\xi R(\xi) \nabla_\theta p_\theta(\xi) \\
&= \sum_\xi R(\xi) p_\theta(\xi) \nabla_\theta \log p_\theta(\xi) \\
&= \mathbb{E}_{\xi \sim p_\theta}[R(\xi) \nabla_\theta \log p_\theta(\xi)]
\end{aligned}$$

  - Allows estimating $\nabla_\theta J_\theta$ using samples $\xi \sim p_\theta$

REINFORCE

- To find $\nabla_\theta J_\theta = \mathbb{E}_{\xi \sim p_\theta}[R(\xi)\, \nabla_\theta \log p_\theta(\xi)]$, sample $\xi \sim p_\theta$, then:

$$\nabla_\theta \log p_\theta(\xi) = \nabla_\theta \left(\log p(s_0) + \sum_t \log \pi_\theta(a_t \,|\, s_t) + \log p(s_{t+1} \,|\, s_t, a_t) \right)$$

$$= \nabla_\theta \sum_t \log \pi_\theta(a_t \,|\, s_t)$$

  - Model-free, but on-policy and high variance (like MC)

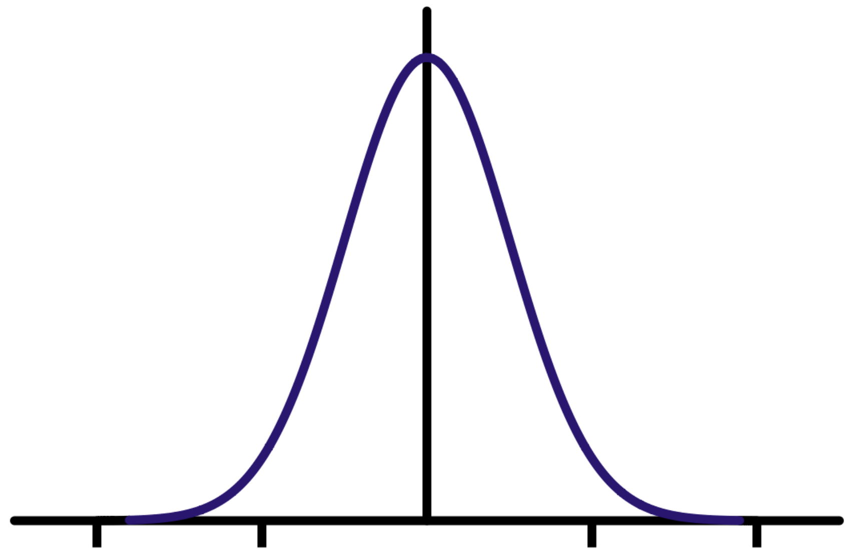
Algorithm REINFORCE

```
Initialize $\pi_\theta$
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

[Williams, 1992]

PG example: Gaussian policy



- How to represent continuous-action policy?
  - One way: Gaussian policy $\pi_\theta(a \mid s) = \mathcal{N}(a; \mu_\theta(s), \Sigma)$
- Log-probability: $\log \pi_\theta(a \mid s) = -\frac{1}{2}\|a - \mu_\theta(s)\|^2_{\Sigma^{-1}} + \text{const}$
  - Where $\|x\|^2_P = x^\intercal P x$ is the Mahalanobis norm
- Policy Gradient:

$$g_\theta(\xi) = R(\xi)\,\nabla_\theta \log p_\theta(\xi) = R(\xi) \sum_t \Sigma^{-1}(a_t - \mu_\theta(s_t))\,\nabla_\theta \mu_\theta(s_t)$$

  - Update $\mu_\theta(s_t)$ toward a_t, more so the higher the return

PG: minimizing reward-surprisal

$$g_\theta(\xi) = R(\xi) \sum_t \nabla_\theta \log \pi_\theta(a_t | s_t)$$

- Surprisal = $-\log \pi_\theta(a | s)$
  - Update θ toward being less surprised by high return
- Surprisal can get very large for unlikely actions
  - Particularly if we try to converge $\pi_\theta(a | s) \to 0$ for suboptimal actions
  - $\Rightarrow$ gradient estimator can have high variance
- Coming up: variance reduction through critics and baselines

Today's lecture

Policy Gradient

Actor–Critic PG

Advantage estimation

Don't let the past distract you

- In $\nabla_\theta J_\theta = \mathbb{E}_{\xi \sim p_\theta}[R(\xi)\,\nabla_\theta \log p_\theta(\xi)]$, both R and $\log p_\theta$ are sums over time

- In finite horizon:

$$\nabla_\theta J_\theta = \nabla_\theta \mathbb{E}_{\xi \sim p_\theta}[R(\xi)] = \sum_{t'=0}^{T-1} \nabla_\theta \mathbb{E}_{\xi_{\leq t'} \sim p_\theta}[r_{t'}] \qquad \textbf{independent of the future}$$

$$= \sum_{t'=0}^{T-1} \mathbb{E}_{\xi_{\leq t'} \sim p_\theta}\left[r_{t'} \sum_{t \leq t'} \nabla_\theta \log \pi_\theta(a_t \,|\, s_t) \right] \qquad \textbf{score-function trick}$$

$$= \sum_{t=0}^{T-1} \mathbb{E}_{\xi \sim p_\theta}[R_{\geq t}(\xi)\,\nabla_\theta \log \pi_\theta(a_t \,|\, s_t)] \qquad \textbf{switch summation order}$$

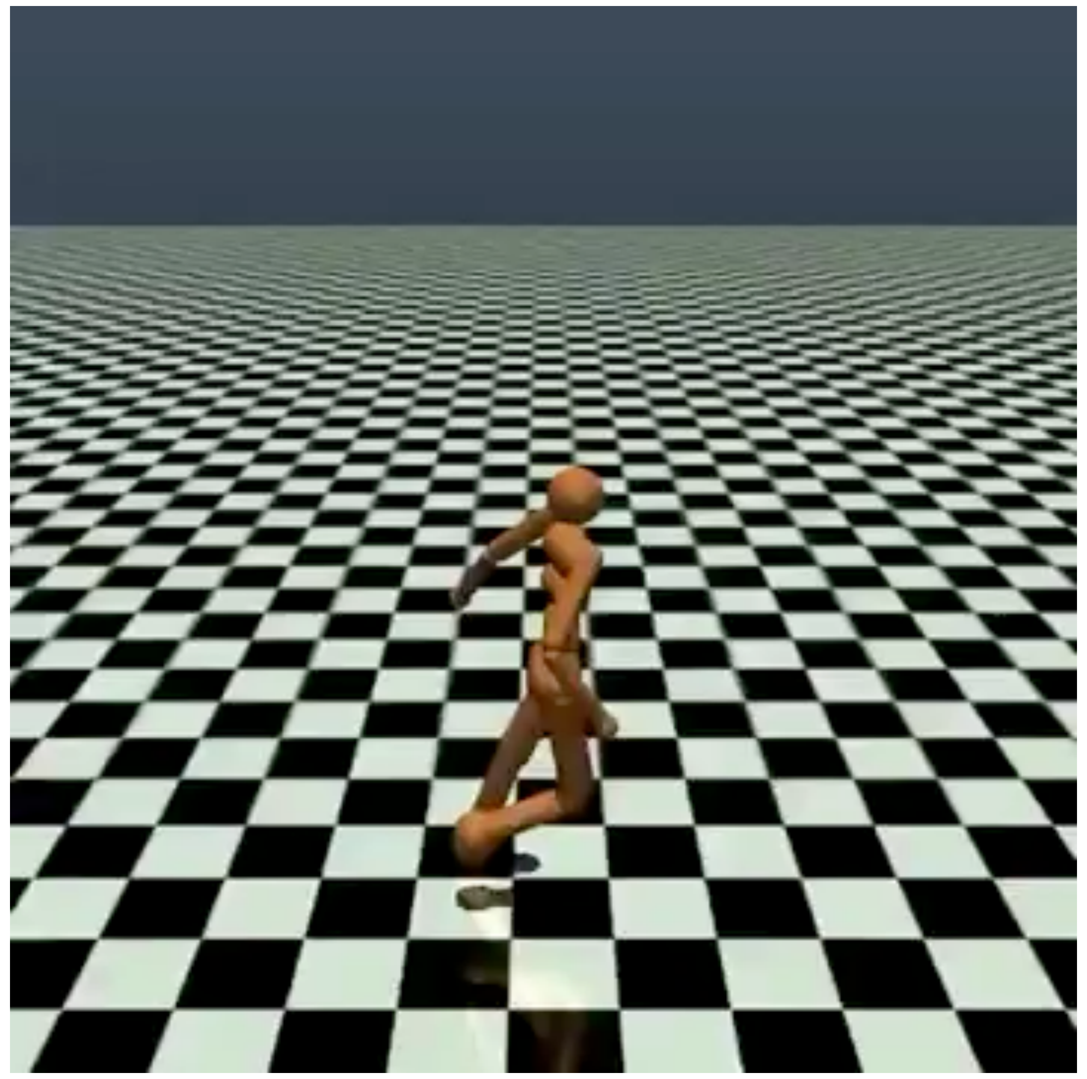
only future return
⇒ less variance

PG with discounted returns

- In discounted horizon:

$$\nabla_\theta J_\theta = \sum_{t \leq t'} \mathbb{E}_{\xi \sim p_\theta}[\gamma^{t'} r_{t'} \nabla_\theta \log \pi_\theta(a_t | s_t)]$$

$$= \sum_t \gamma^t \mathbb{E}_{\xi \sim p_\theta}[R_{\geq t}(\xi) \nabla_\theta \log \pi_\theta(a_t | s_t)]$$



- $R_{\geq t}(\xi) \nabla_\theta \log \pi_\theta(a_t | s_t)$ is discounted by γ^t; should it be?
  - Neglects data from $t \gg 1/(1-\gamma)$; should it?
- If discounting isn't real, just a computational / statistical trick
  - Don't discount by γ^t (most algorithms don't)

Reducing variance through value estimation

- The past in $R(\xi)$ is terms we can't control ⇒ ignore to reduce variance

- The future $R_{\geq t}(\xi)$ is still high-variance ⇒ estimate with TD
  - Replace $R_{\geq t}(\xi)$ with $Q_{\pi_\theta}(s_t, a_t) = \mathbb{E}_{\xi \sim p_\theta}[R_{\geq t}(\xi) \,|\, s_t, a_t]$

- But is it correct?

$$\begin{aligned} \nabla_\theta J_\theta &= \sum_t \gamma^t \mathbb{E}_{\xi \sim p_\theta}[R_{\geq t}(\xi) \nabla_\theta \log \pi_\theta(a_t \,|\, s_t)] \\ &\overset{?}{=} \sum_t \gamma^t \mathbb{E}_{(s_t, a_t) \sim p_\theta}[Q_{\pi_\theta}(s_t, a_t) \nabla_\theta \log \pi_\theta(a_t \,|\, s_t)] \end{aligned}$$

Policy-Gradient Theorem

- Apply chain rule on the value gradient:

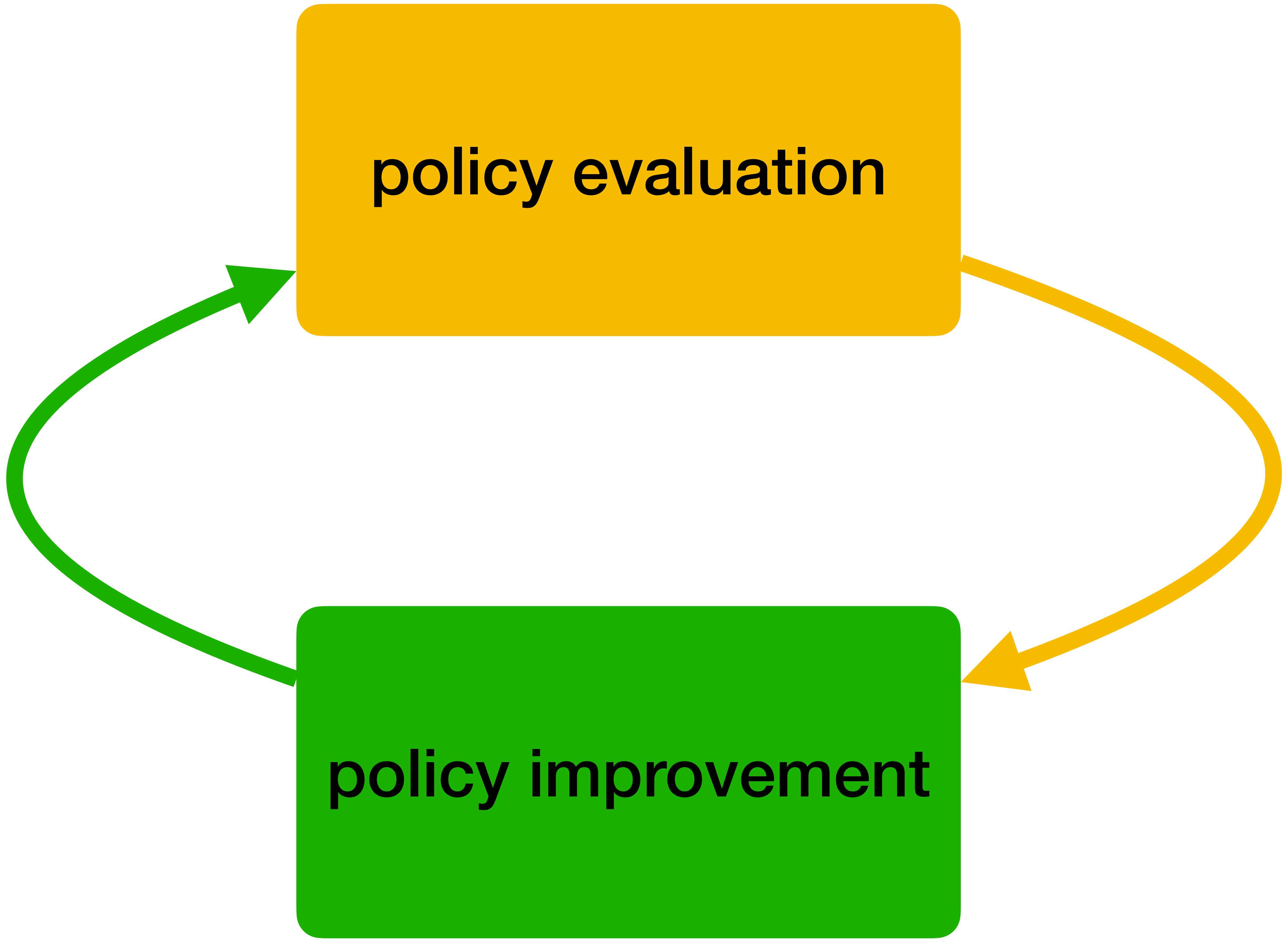
$$\begin{aligned}\nabla_\theta V_{\pi_\theta}(s) &= \nabla_\theta \mathbb{E}_{(a|s)\sim\pi_\theta}[Q_{\pi_\theta}(s,a)] \\ &= \sum_a Q_{\pi_\theta}(s,a)\,\nabla_\theta \pi_\theta(a\,|\,s) + \pi_\theta(a\,|\,s)\,\nabla_\theta Q_{\pi_\theta}(s,a) \qquad \textbf{product rule} \\ &= \mathbb{E}_{(a|s)\sim\pi_\theta}[Q_{\pi_\theta}(s,a)\,\nabla_\theta \log \pi_\theta(a\,|\,s) + \gamma \mathbb{E}_{(s'|s,a)\sim p}[\,\nabla_\theta V_{\pi_\theta}(s')]]\end{aligned}$$

- Here back-propagating gradients is like a Bellman recursion
  - With pseudo-reward $\tilde{r}(s,a) = Q_{\pi_\theta}(s,a)\,\nabla_\theta \log \pi_\theta(a\,|\,s)$

$$\nabla_\theta J_\theta = \sum_t \gamma^t \mathbb{E}_{(s_t,a_t)\sim p_\theta}[\tilde{r}_t(s_t,a_t)] = \sum_t \gamma^t \mathbb{E}_{(s_t,a_t)\sim p_\theta}[Q_{\pi_\theta}(s_t,a_t)\,\nabla_\theta \log \pi_\theta(a_t\,|\,s_t)]$$

[Sutton et al., 2000]

Actor–Critic (AC) methods



policy evaluation

policy improvement

critic $Q_\phi(s, a)$
evaluates actor

actor $\pi_\theta(a \mid s)$
uses critic to improve

Actor–Critic PG

```
Algorithm Actor–Critic PG (Q version)
  Initialize $\pi_\theta$ and $Q_\phi$
  repeat
      Roll out $\xi \sim p_\theta$
      Update $\pi_\theta$ with $g \leftarrow \sum_t Q_\phi(s_t, a_t) \nabla_\theta \log \pi_\theta(a_t | s_t)$
      Update $Q_\phi$ with MC or TD
```

```
Algorithm Actor–Critic PG (V version)
  Initialize $\pi_\theta$ and $V_\phi$
  repeat
      Roll out $\xi \sim p_\theta$
      Update $\pi_\theta$ with $g \leftarrow \sum_t (r_t + \gamma V_\phi(s_{t+1})) \nabla_\theta \log \pi_\theta(a_t | s_t)$
      Update $V_\phi$ with MC or TD
```

MF

θ

DP

π'

max

Today's lecture

Policy Gradient

Actor–Critic PG

Advantage estimation

Baselines

- Constant shift b in return doesn't matter for the policy gradient

$$\mathbb{E}_{\xi \sim p_\theta}[(R(\xi) - b) \nabla_\theta \log p_\theta(\xi)] = \nabla_\theta \mathbb{E}_{\xi \sim p_\theta}[R(\xi) - b] = \nabla_\theta J_\theta$$

$\mathbb{E}[b] = b$ **independent of** θ

- But it can make a huge difference in its variance
  - Consider $\mathbb{E}[xy]$ vs. $\mathbb{E}[x(y + 100)]$, with uniform $(x, y) \in \{-1,1\}^2$
- Making $y - b$ zero-mean (minimum $\mathbb{V}[y - b]$) is a good rule of thumb:
  - Update $b \to R(\xi)$ (approaches the expected return)
  - Estimate $\nabla_\theta J_\theta \approx (R(\xi) - b) \nabla_\theta \log p_\theta(\xi)$

State-dependent baselines

- What can b depend on?

$$\mathbb{E}_{(s_t,a_t)\sim p_\theta}[b\,\nabla_\theta \log \pi_\theta(a_t \mid s_t)] = \mathbb{E}_{s_t\sim p_\theta}[\,\nabla_\theta \mathbb{E}_{(a_t|s_t)\sim\pi_\theta}[b]] = 0$$

  - As long as b is independent of a_t given s_t (i.e. not caused by $\pi_\theta(a_t \mid s_t)$)

- Updating $b(s_t) \to R_{\geq t}(\xi) \Rightarrow$ we're learning V_{π_θ}

- In the TD PG version:

$$\nabla_\theta J_\theta = \sum_t \gamma^t \mathbb{E}_{(s_t,a_t)\sim p_\theta}[(Q_{\pi_\theta}(s_t, a_t) - V_{\pi_\theta}(s_t))\,\nabla_\theta \log \pi_\theta(a_t \mid s_t)]$$

Advantage estimation

- Advantage function: $A_\pi(s,a) = Q_\pi(s,a) - V_\pi(s)$
- AC PG with baseline: $\nabla_\theta J_\theta = \sum_t \gamma^t \mathbb{E}_{(s_t,a_t)\sim p_\theta}[A_{\pi_\theta}(s_t,a_t)\nabla_\theta \log \pi_\theta(a_t|s_t)]$
- How to estimate $A(s,a)$ using a critic $V_\phi(s)$?
  - MC:

$$A(s_t,a_t) \approx R_{\geq t}(\xi) - V_\phi(s)$$

  - TD:

$$A(s,a) \approx r + \gamma V_\phi(s') - V_\phi(s)$$

Advantage Actor–Critic (A2C)

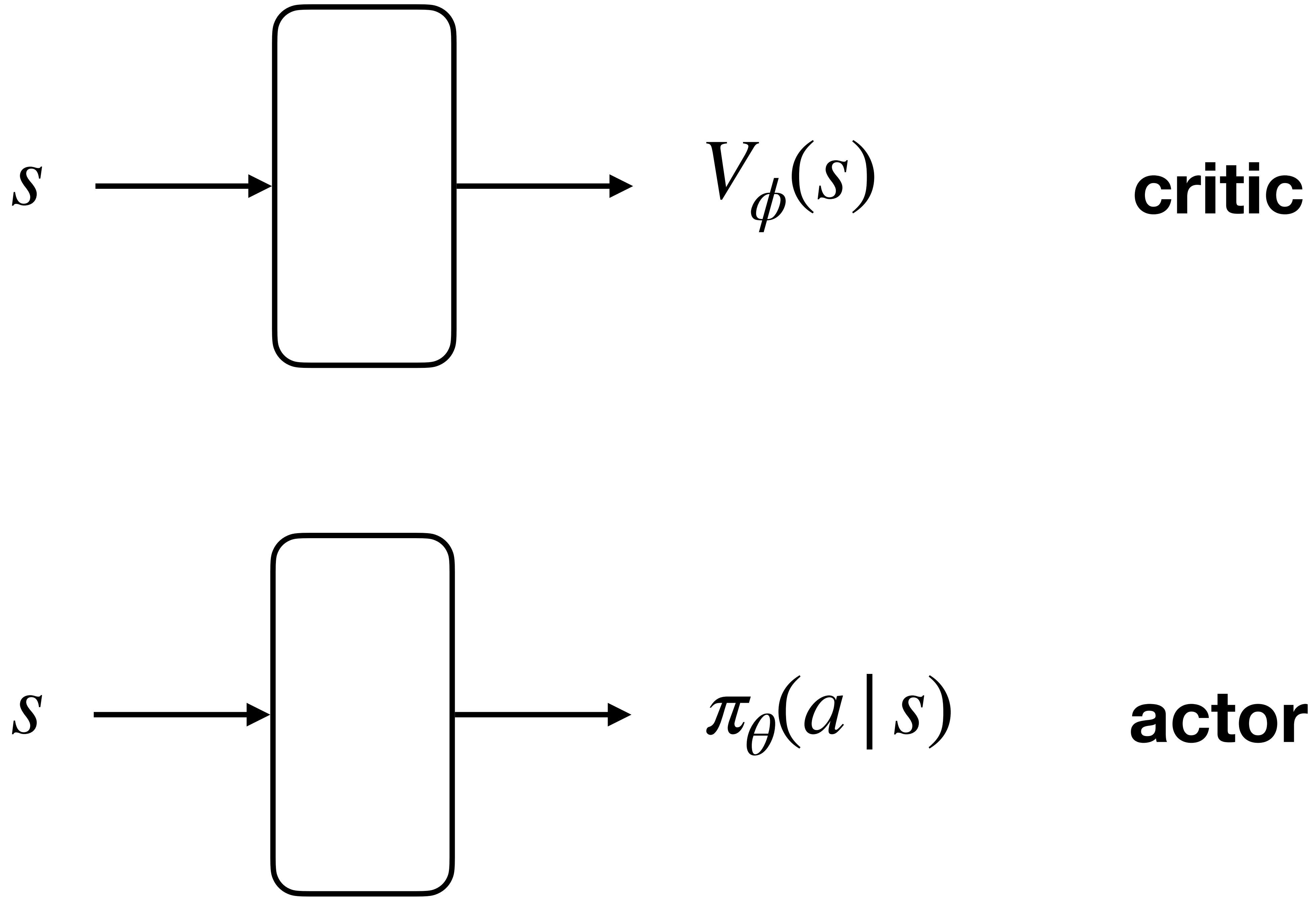
MF
θ
DP
π'
max

```
Algorithm Advantage Actor–Critic
    Initialize $\pi_\theta$ and $V_\phi$
    repeat
        Roll out $\xi \sim p_\theta$
        Update $\Delta\theta \leftarrow \sum_t (R_{\geq t}(\xi) - V_\phi(s_t)) \nabla_\theta \log \pi_\theta(a_t|s_t)$
        Descend $L_\phi = \sum_t (R_{\geq t}(\xi) - V_\phi(s_t))^2$
```

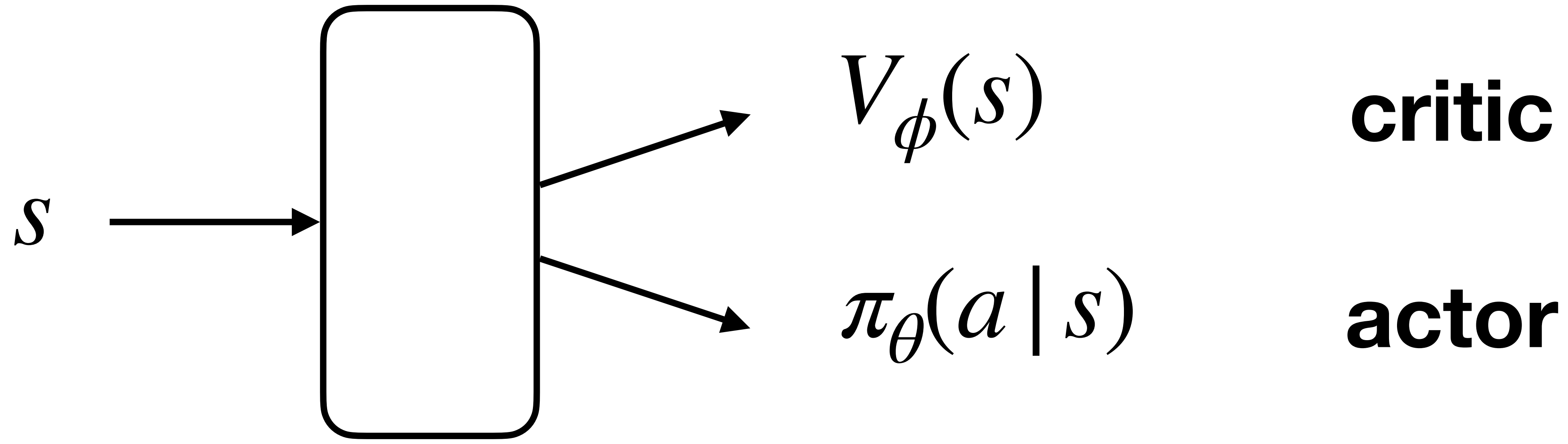
[Mnih et al., 2016]

Practical considerations: param sharing

- Separate parameters:

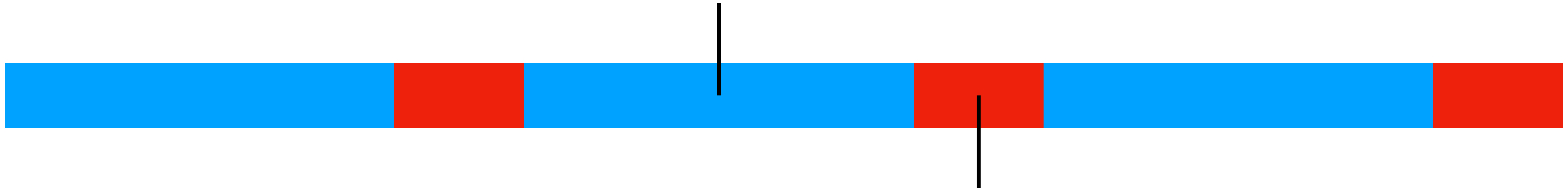


- Shared parameters:
  - Can be more data efficient
  - Can be less stable



Practical considerations: distributed comp.

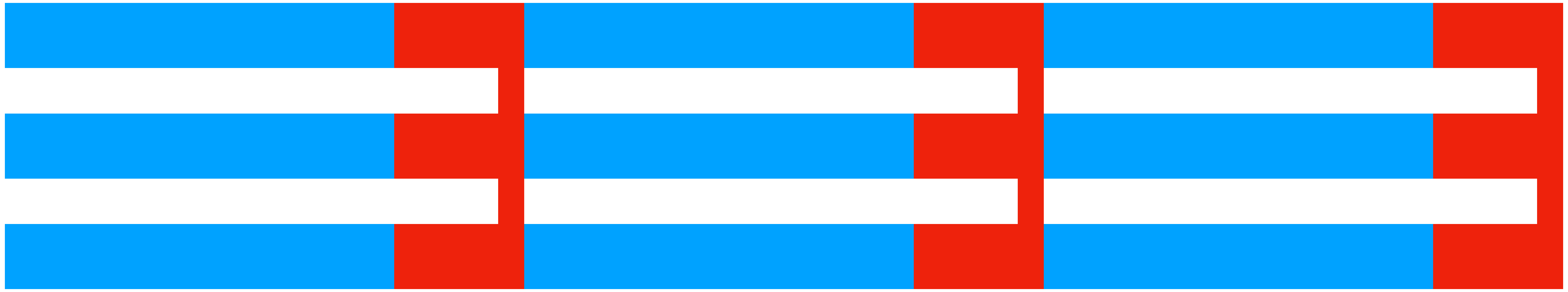
- Serial execution



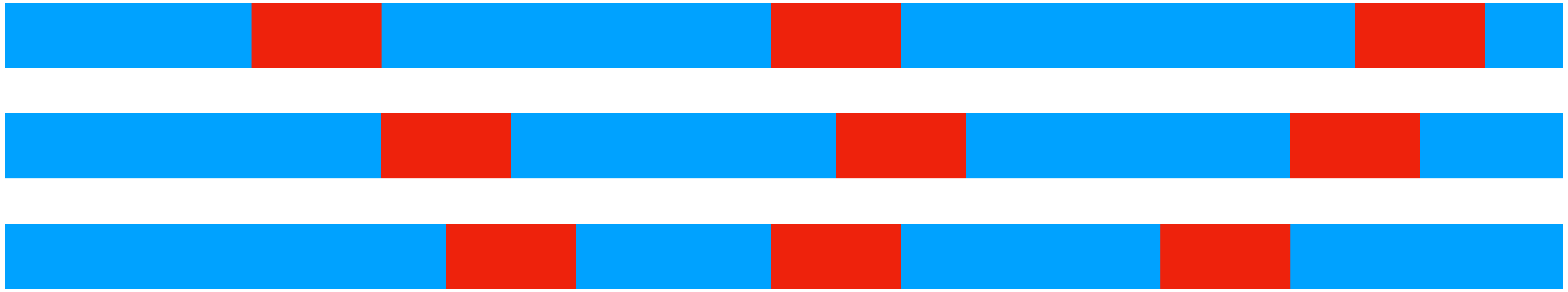
simulate to collect data

take gradient step

- Synchronous parallel execution



- Asynchronous parallel execution (A3C)



[Mnih et al., 2016]

Comparing advantage estimators

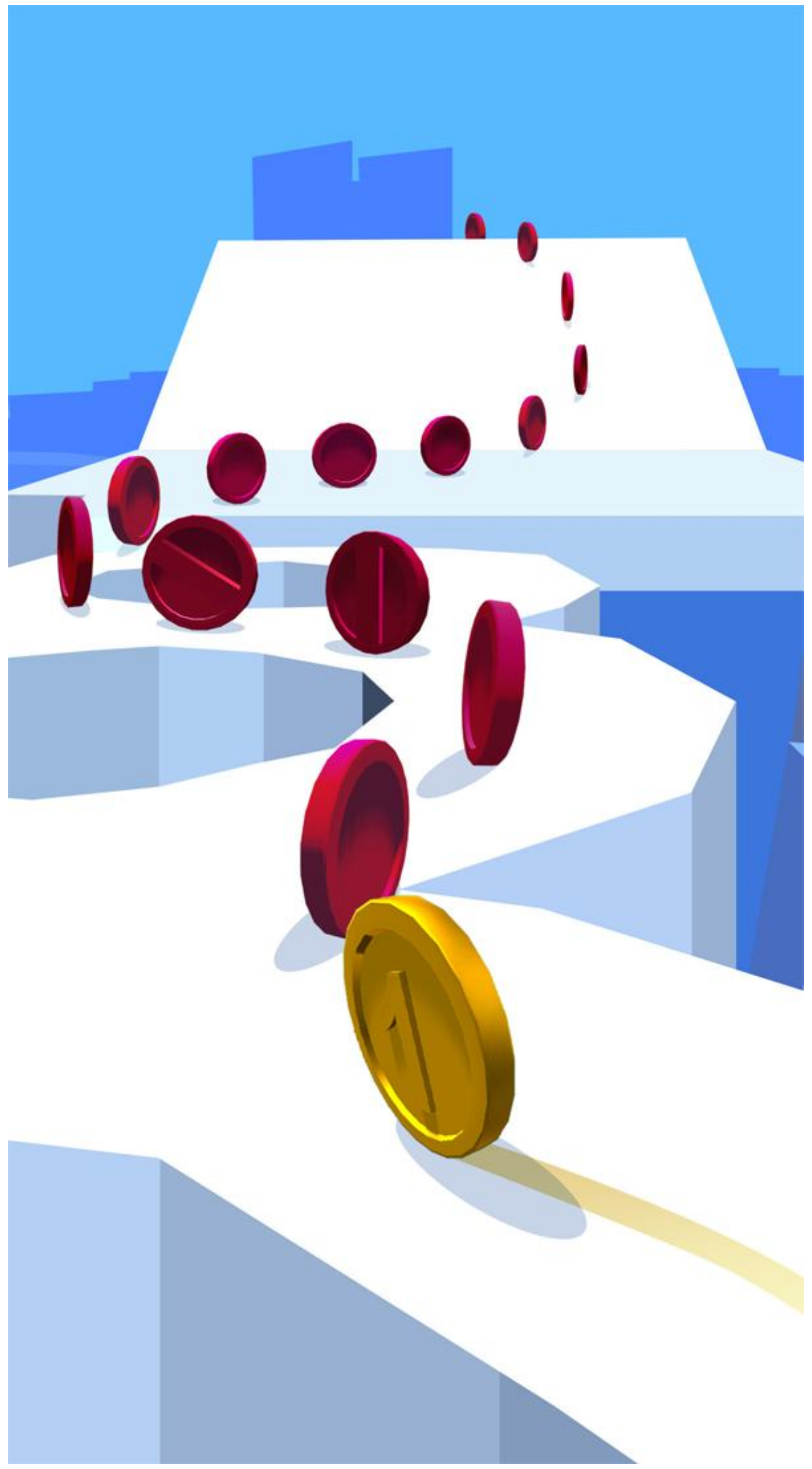
	bias	variance
- Constant baseline $\nabla_\theta J_\theta \approx (R_{\geq t} - b) \nabla_\theta \log \pi_\theta(a_t \mid s_t)$	**none**	**high** **one gradient per trajectory**
- State-based baseline (MC) $\nabla_\theta J_\theta \approx (R_{\geq t} - V_\phi(s_t)) \nabla_\theta \log \pi_\theta(a_t \mid s_t)$	**none**	**mid** **state-dependent baseline**
- State-based baseline (TD) $\nabla_\theta J_\theta \approx (r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)) \nabla_\theta \log \pi_\theta(a_t \mid s_t)$	**some** V_ϕ **is approximate**	**lower**

Multi-step TD

- 1-step TD: $A_t^1 = r_t + \gamma V(s_{t+1}) - V(s_t)$
- 2-step TD: $A_t^2 = r_t + \gamma r_{t+1} + \gamma^2 V(s_{t+2}) - V(s_t)$

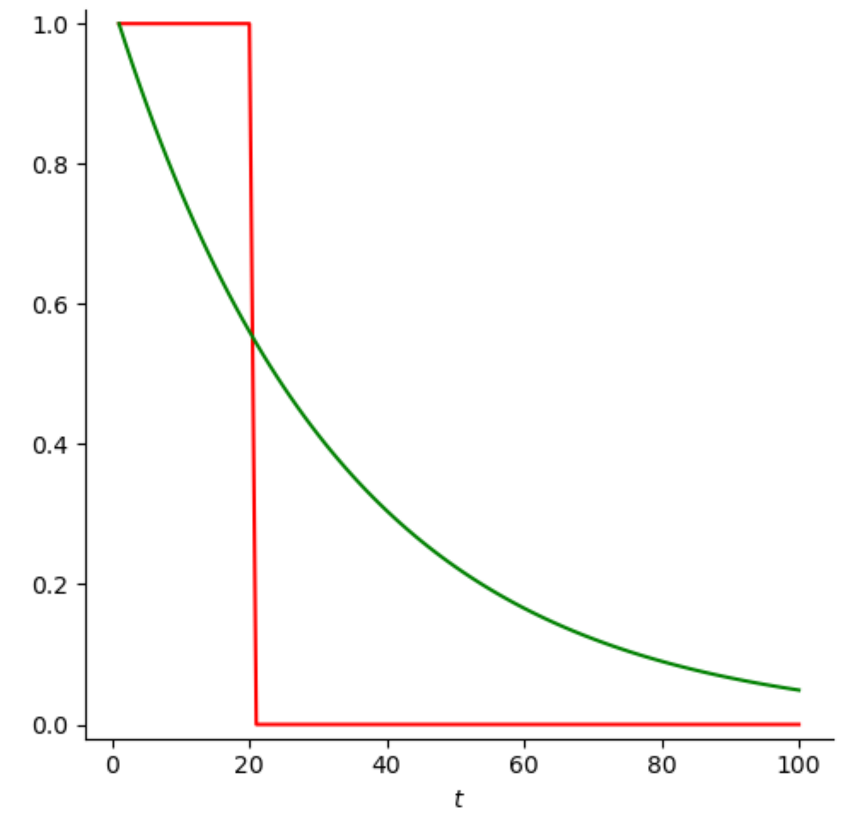
  …

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



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 $0 \leq \lambda \leq 1$



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

$A^1_{t+\Delta t}$

- 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

[Schulman et al., 2015]

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