

Reducing Variance

- Causality trick
- Discount factor
- Baseline
- Actor-critic
- Optimization techniques:
 - Natural gradient
 - Trust region

Reducing Variance: Causality

$$\nabla_{\theta} J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \left(\sum_{t=1}^T \underbrace{\nabla_{\theta} \log \pi_{\theta}(\mathbf{a}_{i,t} | \mathbf{s}_{i,t})}_{\text{prob. traj}} \right) \underbrace{\left(\sum_{t=1}^T r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) \right)}_{\text{return}}$$

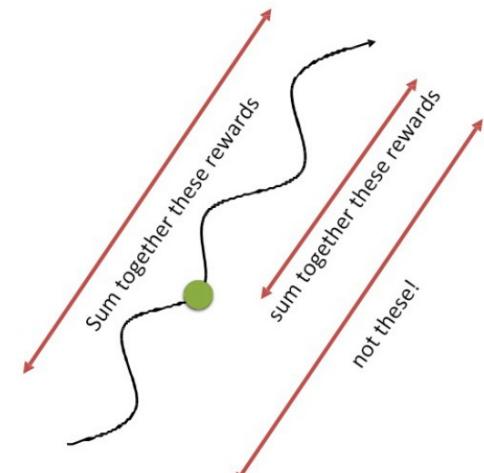
Causality: policy at time t' cannot affect reward at time t when $t < t'$

Original

$$\frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) \left(\sum_{t'=1}^T r(\mathbf{a}_{i,t'}, \mathbf{s}_{i,t'}) \right)$$

$$\frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) \left(\sum_{t'=t}^T r(\mathbf{a}_{i,t'}, \mathbf{s}_{i,t'}) \right)$$

$\stackrel{\text{"reward to go"} \hat{r}(\mathbf{a}_{i,t}, \mathbf{s}_{i,t'})}{=} \mathbb{E} \left(\nabla_{\theta} \log \pi_{\theta}(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) \sum_{t'=1}^{t-1} \hat{r}(\mathbf{a}_{i,t'}, \mathbf{s}_{i,t'}) \right)$



Reducing Variance: Discount Factor

option 1: $\nabla_{\theta} J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) \left(\sum_{t'=t}^T \gamma^{t'-t} r(\mathbf{s}_{i,t'}, \mathbf{a}_{i,t'}) \right)$

option 2: $\nabla_{\theta} J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \left(\sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) \right) \left(\sum_{t=1}^T \gamma^{t-1} r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) \right)$

Not the same

$\nabla_{\theta} J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) \left(\sum_{t=1}^T \gamma^{t-1} r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) \right)$

$\nabla_{\theta} J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \gamma^{t-1} \nabla_{\theta} \log \pi_{\theta}(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) \left(\sum_{t'=t}^T \gamma^{t'-t} r(\mathbf{s}_{i,t'}, \mathbf{a}_{i,t'}) \right)$

Reducing Variance: Baselines

$$\nabla_{\theta} J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \nabla_{\theta} \log p_{\theta}(\tau) [r(\tau) - b]$$

$$b = \frac{1}{N} \sum_{i=1}^N r(\tau)$$

$$E[\nabla_{\theta} \log p_{\theta}(\tau) b] = \int p_{\theta}(\tau) \nabla_{\theta} \log p_{\theta}(\tau) b d\tau = \int \nabla_{\theta} p_{\theta}(\tau) b d\tau = b \nabla_{\theta} \underbrace{\int p_{\theta}(\tau) d\tau}_1 = b \nabla_{\theta} 1 = 0$$

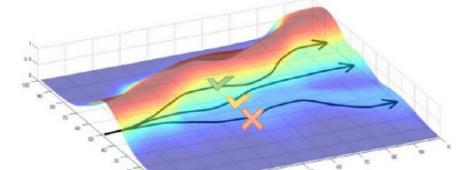
const(τ)

subtracting a baseline is *unbiased* in expectation!

average reward is *not* the best baseline, but it's pretty good!

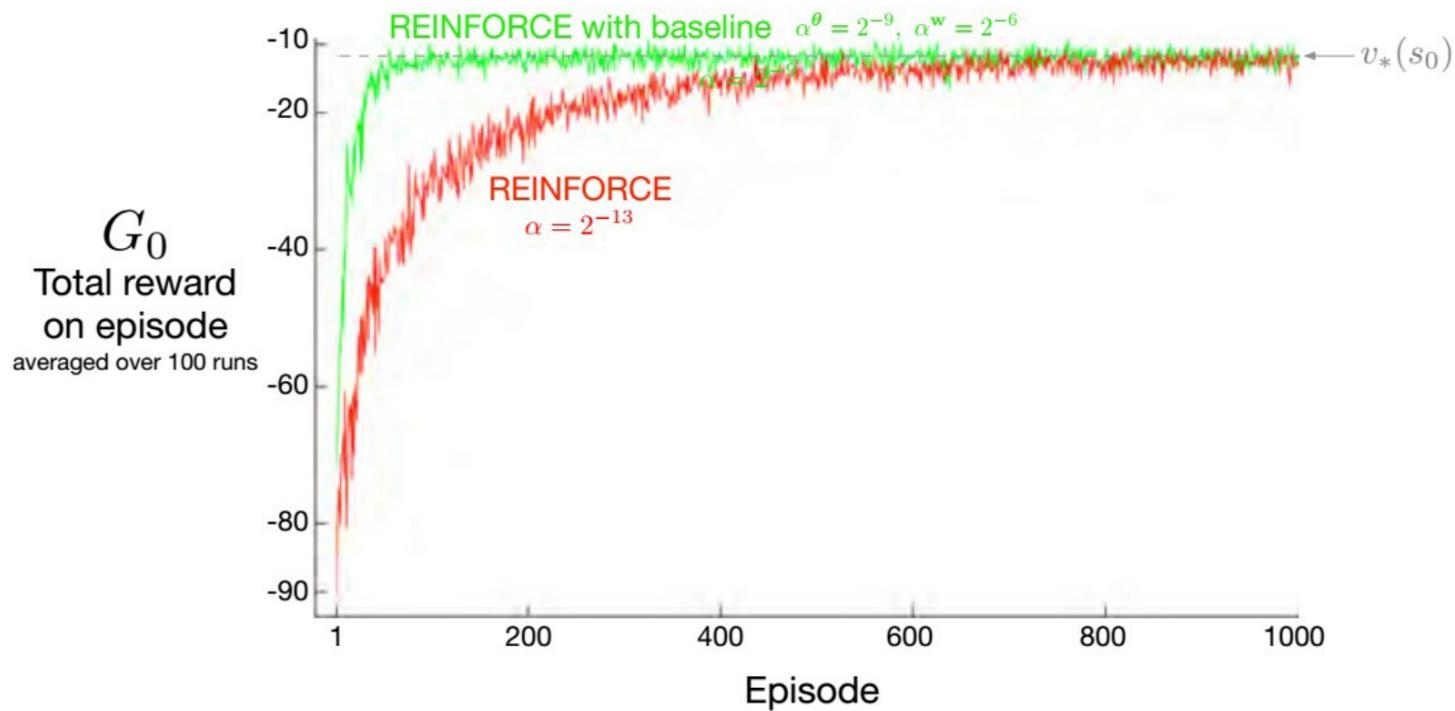
a convenient identity

$$p_{\theta}(\tau) \nabla_{\theta} \log p_{\theta}(\tau) = \nabla_{\theta} p_{\theta}(\tau)$$



Reducing Variance: Baselines

Faster convergence:



Analyzing Variance

$$\min \text{Var}[f(b, \dots, b)]$$

$$\min \text{Var}[f(b_1, \dots, b_n)]$$

$$\text{Var}[x] = E[x^2] - E[x]^2$$

$$\nabla_{\theta} J(\theta) = E_{\tau \sim p_{\theta}(\tau)} [\nabla_{\theta} \log p_{\theta}(\tau) (r(\tau) - b)]$$

$$\text{Var} = E_{\tau \sim p_{\theta}(\tau)} [(\underbrace{\nabla_{\theta} \log p_{\theta}(\tau) (r(\tau) - b)}_{g(\tau)})^2] - E_{\tau \sim p_{\theta}(\tau)} [\nabla_{\theta} \log p_{\theta}(\tau) (r(\tau) - b)]^2$$

const(b)

this bit is just $E_{\tau \sim p_{\theta}(\tau)} [\nabla_{\theta} \log p_{\theta}(\tau) r(\tau)]$
 (baselines are unbiased in expectation)

$$\begin{aligned} \frac{d \text{Var}}{db} &= \frac{d}{db} E[g(\tau)^2 (r(\tau) - b)^2] = \frac{d}{db} (E[\cancel{g(\tau)^2 r(\tau)^2}] - 2E[g(\tau)^2 r(\tau)b] + b^2 E[g(\tau)^2]) \\ &= -2E[g(\tau)^2 r(\tau)] + 2bE[g(\tau)^2] = 0 \end{aligned}$$

$$b = \frac{E[g(\tau)^2 r(\tau)]}{E[g(\tau)^2]}$$

This is just expected reward, but weighted by gradient magnitudes!

Reducing Variance: Review

- • Exploiting causality
 - Future doesn't affect the past
- • Discount factor
 - Two different version
- • Baselines
 - Analyzing variance for deriving optimal baselines
 - Now: Introducing actor-critic methods!

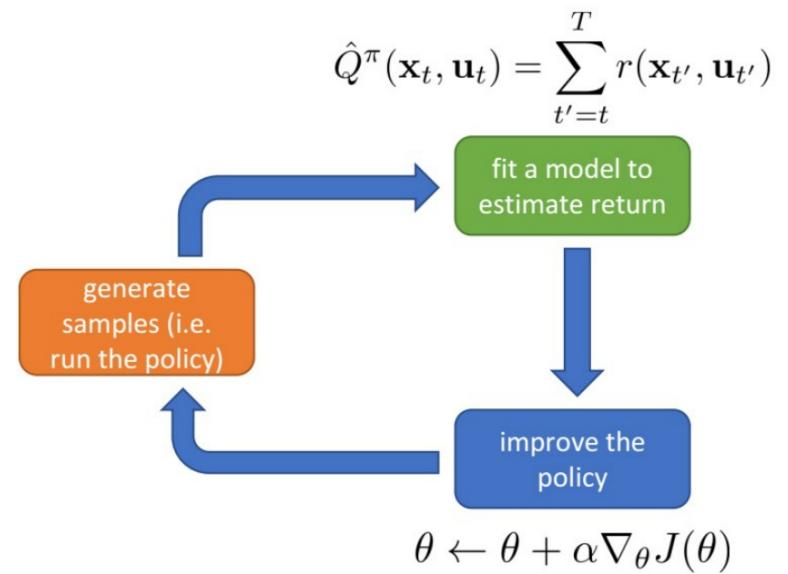
Policy Gradients so Far

REINFORCE algorithm:

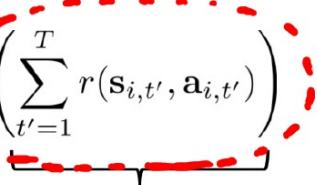
1. sample $\{\tau^i\}$ from $\pi_\theta(\mathbf{a}_t | \mathbf{s}_t)$ (run the policy)
 2. $\nabla_\theta J(\theta) \approx \sum_i \left(\sum_{t=1}^T \nabla_\theta \log \pi_\theta(\mathbf{a}_t^i | \mathbf{s}_t^i) \left(\sum_{t'=t}^T r(\mathbf{s}_{t'}, \mathbf{a}_{t'}^i) \right) \right)$
 3. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$
- $Q^\pi(s, a)$
-

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) \hat{Q}_{i,t}^\pi$$

“reward to go”

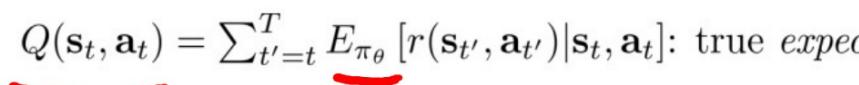


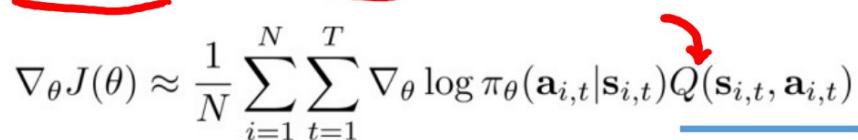
Improving Estimation of Reward to Go

$$\nabla_{\theta} J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) \left(\sum_{t'=1}^T r(\mathbf{s}_{i,t'}, \mathbf{a}_{i,t'}) \right)$$


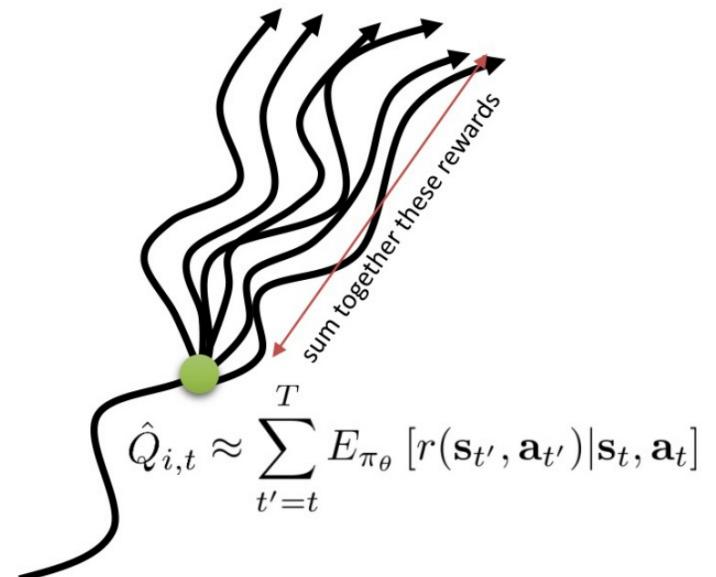
$\hat{Q}_{i,t}$: estimate of expected reward if we take action $\mathbf{a}_{i,t}$ in state $\mathbf{s}_{i,t}$

How to make a better estimate?

$$Q(\mathbf{s}_t, \mathbf{a}_t) = \sum_{t'=t}^T E_{\pi_{\theta}} [r(\mathbf{s}_{t'}, \mathbf{a}_{t'}) | \mathbf{s}_t, \mathbf{a}_t]: \text{true expected reward-to-go}$$


$$\nabla_{\theta} J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) \underline{Q(\mathbf{s}_{i,t}, \mathbf{a}_{i,t})}$$


much lower variance!



Improving Estimation of Reward to Go

Further improvement: Adding a baseline!

$Q(\mathbf{s}_t, \mathbf{a}_t) = \sum_{t'=t}^T E_{\pi_\theta} [r(\mathbf{s}_{t'}, \mathbf{a}_{t'}) | \mathbf{s}_t, \mathbf{a}_t]$: true expected reward-to-go

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) (Q(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) - b_t)$$

baseline

$$b_t = \frac{1}{N} \sum_i Q(\mathbf{s}_{i,t}, \mathbf{a}_{i,t})$$

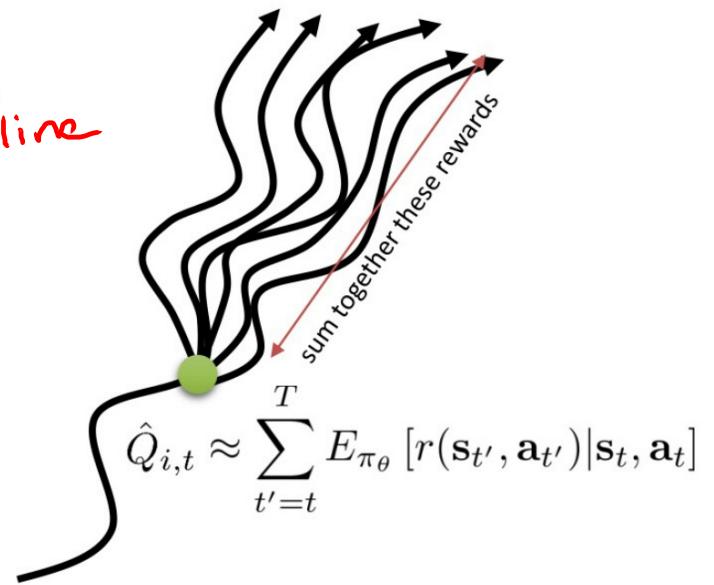
$a_{i,t} \sim \pi_\theta$

$\overbrace{\quad}^{\text{return}}$

\downarrow

$\overbrace{\quad}^{\text{value}}$

$V(s_{i,t})$



Improving Estimation of Reward to Go

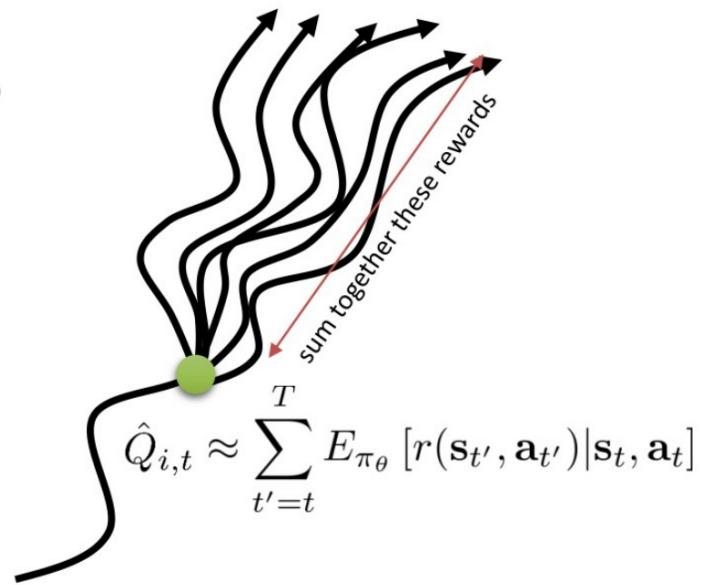
Further improvement: Adding a baseline!

$$Q(\mathbf{s}_t, \mathbf{a}_t) = \sum_{t'=t}^T E_{\pi_\theta} [r(\mathbf{s}_{t'}, \mathbf{a}_{t'}) | \mathbf{s}_t, \mathbf{a}_t]: \text{true expected reward-to-go}$$

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) (Q(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) - V(\mathbf{s}_{i,t}))$$

$$b_t = \frac{1}{N} \sum_i Q(\mathbf{s}_{i,t}, \mathbf{a}_{i,t})$$

$$V(\mathbf{s}_t) = E_{\mathbf{a}_t \sim \pi_\theta(\mathbf{a}_t | \mathbf{s}_t)} [Q(\mathbf{s}_t, \mathbf{a}_t)]$$



Advantage Value

$Q^\pi(\mathbf{s}_t, \mathbf{a}_t) = \sum_{t'=t}^T E_{\pi_\theta}[r(\mathbf{s}_{t'}, \mathbf{a}_{t'}) | \mathbf{s}_t, \mathbf{a}_t]$: total reward from taking \mathbf{a}_t in \mathbf{s}_t

$V^\pi(\mathbf{s}_t) = E_{\mathbf{a}_t \sim \pi_\theta(\mathbf{a}_t | \mathbf{s}_t)}[Q^\pi(\mathbf{s}_t, \mathbf{a}_t)]$: total reward from \mathbf{s}_t

$A^\pi(\mathbf{s}_t, \mathbf{a}_t) = \underbrace{Q^\pi(\mathbf{s}_t, \mathbf{a}_t)}_{\text{Reward}} - \underbrace{V^\pi(\mathbf{s}_t)}_{\text{baseline}}$: how much better \mathbf{a}_t is

to go

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) A^\pi(\mathbf{s}_{i,t}, \mathbf{a}_{i,t})$$



the better this estimate, the lower the variance

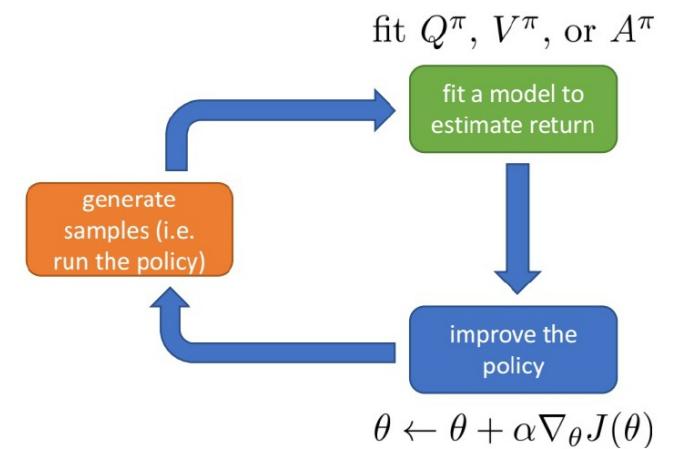
Advantage Value Approximation

$$Q^\pi(\mathbf{s}_t, \mathbf{a}_t) = \sum_{t'=t}^T E_{\pi_\theta}[r(\mathbf{s}_{t'}, \mathbf{a}_{t'}) | \mathbf{s}_t, \mathbf{a}_t]$$

$$V^\pi(\mathbf{s}_t) = E_{\mathbf{a}_t \sim \pi_\theta(\mathbf{a}_t | \mathbf{s}_t)}[Q^\pi(\mathbf{s}_t, \mathbf{a}_t)]$$

$$A^\pi(\mathbf{s}_t, \mathbf{a}_t) = Q^\pi(\mathbf{s}_t, \mathbf{a}_t) - V^\pi(\mathbf{s}_t)$$

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) A^\pi(\mathbf{s}_{i,t}, \mathbf{a}_{i,t})$$



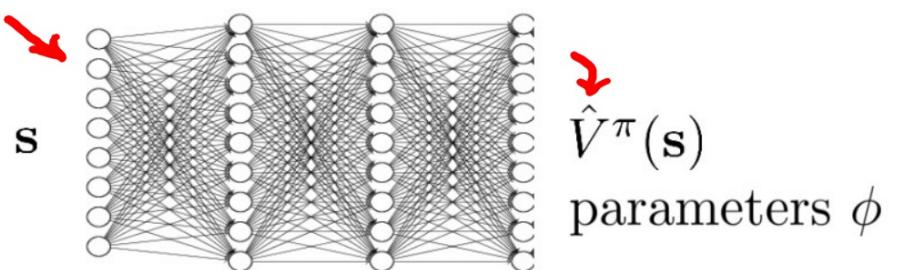
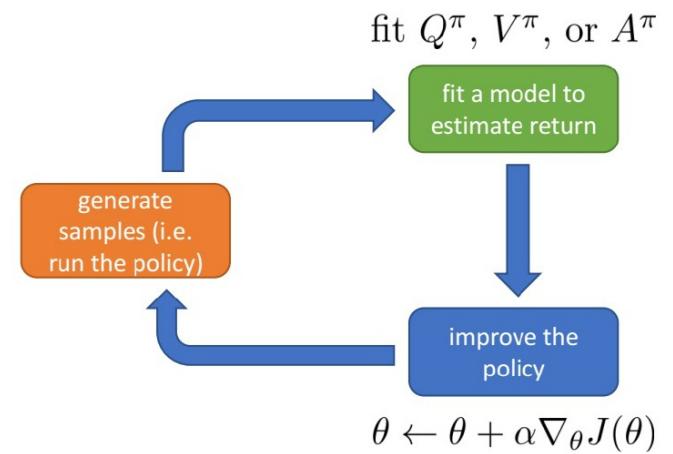
$$\begin{aligned}
 Q^\pi(\mathbf{s}_t, \mathbf{a}_t) &= \sum_{t'=t}^T E_{\pi_\theta}[r(\mathbf{s}_{t'}, \mathbf{a}_{t'}) | \mathbf{s}_t, \mathbf{a}_t] \\
 &= r(\mathbf{s}_t, \mathbf{a}_t) + \sum_{t'=t+1}^T E_{\pi_\theta}[r(\mathbf{s}_{t'}, \mathbf{a}_{t'}) | \mathbf{s}_t, \mathbf{a}_t] \\
 &\approx V^\pi(\mathbf{s}_{t+1})
 \end{aligned}$$

Advantage Value Approximation

$$Q^\pi(\mathbf{s}_t, \mathbf{a}_t) \approx r(\mathbf{s}_t, \mathbf{a}_t) + V^\pi(\mathbf{s}_{t+1})$$

$$A^\pi(\mathbf{s}_t, \mathbf{a}_t) \approx r(\mathbf{s}_t, \mathbf{a}_t) + V^\pi(\mathbf{s}_{t+1}) - V^\pi(\mathbf{s}_t)$$

let's just fit $V^\pi(\mathbf{s})$!



Policy Evaluation

$$V^\pi(\mathbf{s}_t) = \sum_{t'=t}^T E_{\pi_\theta} [r(\mathbf{s}_{t'}, \mathbf{a}_{t'}) | \mathbf{s}_t]$$

$$J(\theta) = E_{\mathbf{s}_1 \sim p(\mathbf{s}_1)} [V^\pi(\mathbf{s}_1)]$$

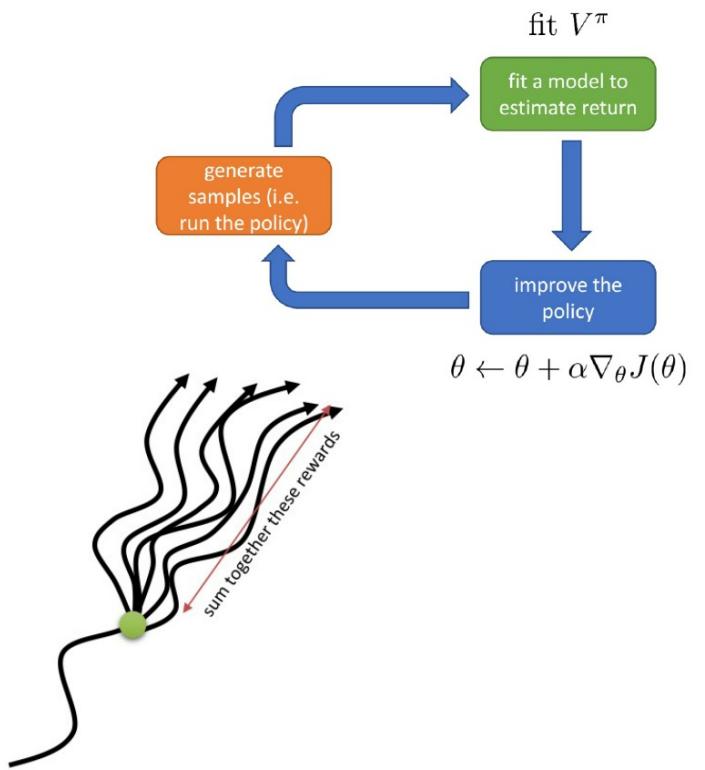
how can we perform policy evaluation?

Monte Carlo policy evaluation (this is what policy gradient does)

$$V^\pi(\mathbf{s}_t) \approx \sum_{t'=t}^T r(\mathbf{s}_{t'}, \mathbf{a}_{t'})$$

$$V^\pi(\mathbf{s}_t) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t'=t}^T r(\mathbf{s}_{t'}, \mathbf{a}_{t'})$$

(requires us to reset the simulator)



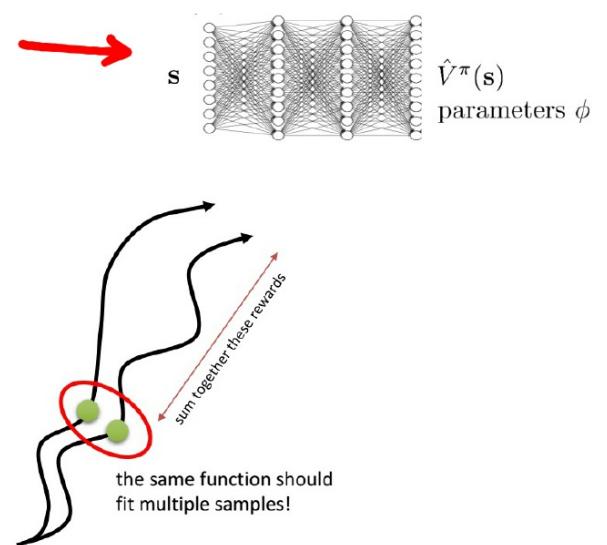
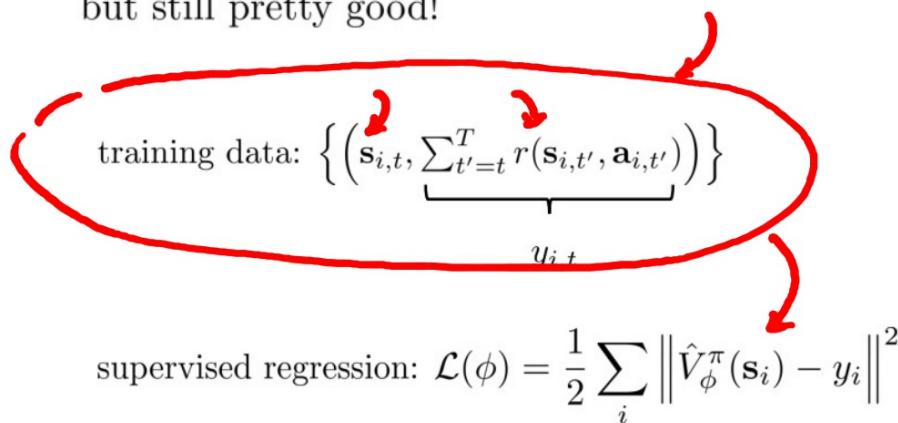
Policy Evaluation

Monte Carlo estimation with function approximator:

$$V^\pi(\mathbf{s}_t) \approx \sum_{t'=t}^T r(\mathbf{s}_{t'}, \mathbf{a}_{t'})$$

not as good as this: $V^\pi(\mathbf{s}_t) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t'=t}^T r(\mathbf{s}_{t'}, \mathbf{a}_{t'})$

but still pretty good!



Policy Evaluation

How to make a better estimate?

$$\text{ideal target: } y_{i,t} = \sum_{t'=t}^T E_{\pi_\theta} [r(\mathbf{s}_{t'}, \mathbf{a}_{t'}) | \mathbf{s}_{i,t}] \approx r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) + V^\pi(\mathbf{s}_{i,t+1}) \approx r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) + \underbrace{\hat{V}_\phi^\pi(\mathbf{s}_{i,t+1})}$$

$$\text{Monte Carlo target: } y_{i,t} = \sum_{t'=t}^T r(\mathbf{s}_{i,t'}, \mathbf{a}_{i,t'})$$

directly use previous fitted value function!

Policy Evaluation

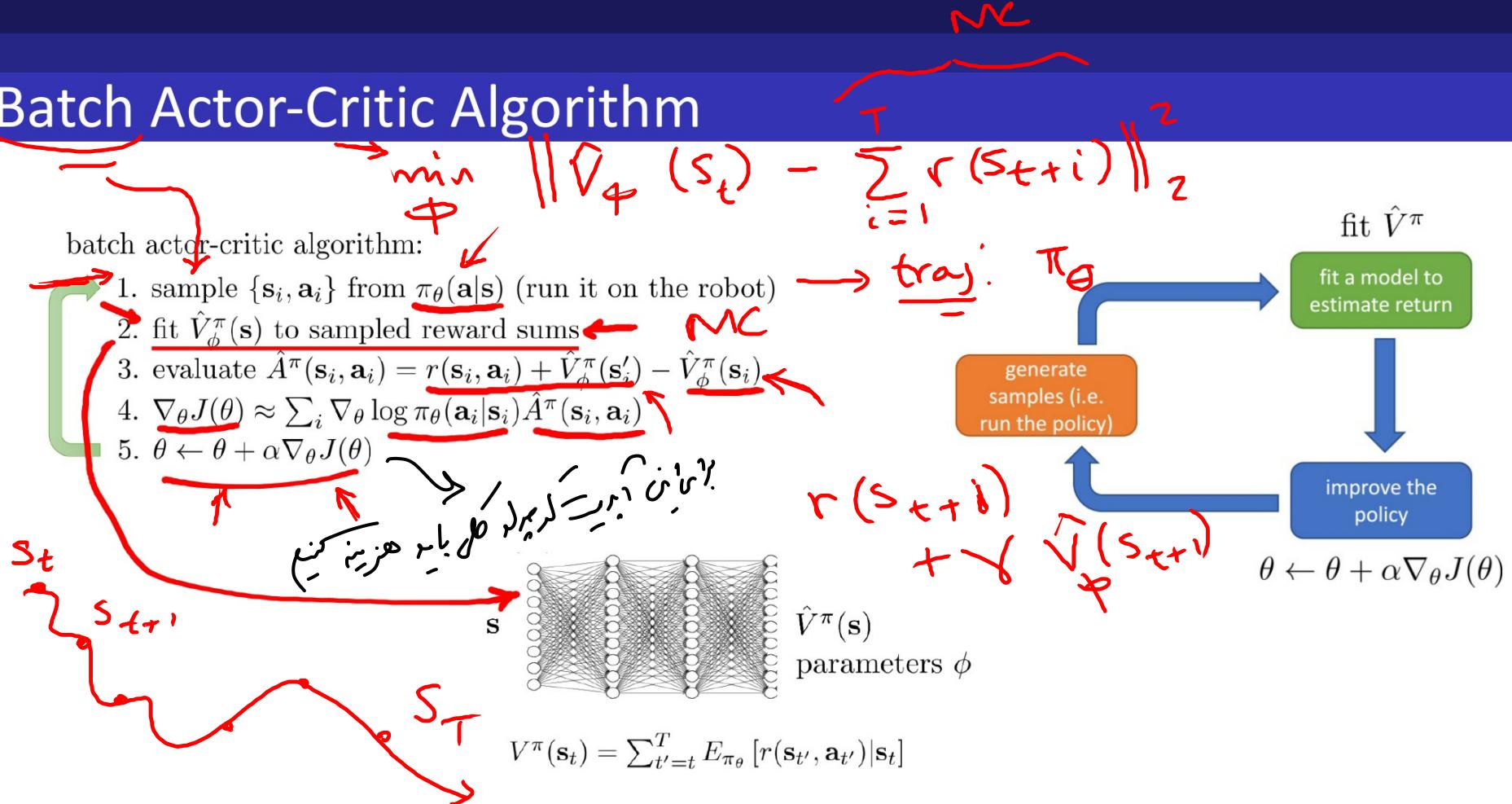
Bootstrap Estimation with Function Approximator

ideal target: $y_{i,t} = \sum_{t'=t}^T E_{\pi_\theta} [r(\mathbf{s}_{t'}, \mathbf{a}_{t'}) | \mathbf{s}_{i,t}] \approx r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) + \hat{V}_\phi^\pi(\mathbf{s}_{i,t+1})$

training data: $\left\{ \left(\mathbf{s}_{i,t}, \underbrace{r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) + \hat{V}_\phi^\pi(\mathbf{s}_{i,t+1})}_{y_{i,t}} \right) \right\}$

supervised regression: $\mathcal{L}(\phi) = \frac{1}{2} \sum_i \left\| \hat{V}_\phi^\pi(\mathbf{s}_i) - y_i \right\|^2$

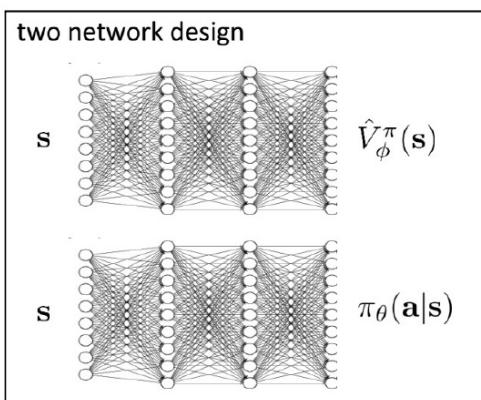
Batch Actor-Critic Algorithm



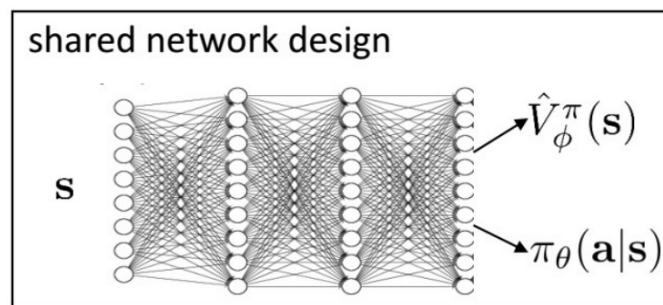
Actor-Critic Algorithm: Architecture Design

batch actor-critic algorithm:

1. sample $\{\mathbf{s}_i, \mathbf{a}_i\}$ from $\pi_\theta(\mathbf{a}|\mathbf{s})$ (run it on the robot)
2. fit $\hat{V}_\phi^\pi(\mathbf{s})$ to sampled reward sums
3. evaluate $\hat{A}^\pi(\mathbf{s}_i, \mathbf{a}_i) = r(\mathbf{s}_i, \mathbf{a}_i) + \hat{V}_\phi^\pi(\mathbf{s}'_i) - \hat{V}_\phi^\pi(\mathbf{s}_i)$
4. $\nabla_\theta J(\theta) \approx \sum_i \nabla_\theta \log \pi_\theta(\mathbf{a}_i|\mathbf{s}_i) \hat{A}^\pi(\mathbf{s}_i, \mathbf{a}_i)$
5. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$



+ simple & stable
- no shared features between actor & critic



Policy Evaluation in Infinite Horizon Settings

$$y_{i,t} \approx r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) + \hat{V}_\phi^\pi(\mathbf{s}_{i,t+1})$$

$$\mathcal{L}(\phi) = \frac{1}{2} \sum_i \left\| \hat{V}_\phi^\pi(\mathbf{s}_i) - y_i \right\|^2$$

what if T (episode length) is ∞ ?

\hat{V}_ϕ^π can get infinitely large in many cases

simple trick: better to get rewards sooner than later

$$y_{i,t} \approx r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) + \gamma \hat{V}_\phi^\pi(\mathbf{s}_{i,t+1})$$

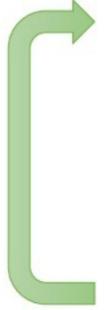
$$\mathcal{L}(\phi) = \frac{1}{2} \sum_i \left\| \hat{V}_\phi^\pi(\mathbf{s}_i) - y_i \right\|^2$$

$$\hat{A}^\pi(\mathbf{s}_{i,t}, \mathbf{a}_{i,t})$$

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) \left(\overbrace{r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) + \gamma \hat{V}_\phi^\pi(\mathbf{s}_{i,t+1}) - \hat{V}_\phi^\pi(\mathbf{s}_{i,t})}^{\text{TD error}} \right)$$

Actor-Critic Algorithm: Infinite Horizon

batch actor-critic algorithm:

- 
1. sample $\{\mathbf{s}_i, \mathbf{a}_i\}$ from $\pi_\theta(\mathbf{a}|\mathbf{s})$ (run it on the robot)
 2. fit $\hat{V}_\phi^\pi(\mathbf{s})$ to sampled reward sums
 3. evaluate $\hat{A}^\pi(\mathbf{s}_i, \mathbf{a}_i) = r(\mathbf{s}_i, \mathbf{a}_i) + \gamma \hat{V}_\phi^\pi(\mathbf{s}'_i) - \hat{V}_\phi^\pi(\mathbf{s}_i)$
 4. $\nabla_\theta J(\theta) \approx \sum_i \nabla_\theta \log \pi_\theta(\mathbf{a}_i|\mathbf{s}_i) \hat{A}^\pi(\mathbf{s}_i, \mathbf{a}_i)$
 5. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

Critics as Baselines

Actor-critic:

$$\nabla_{\theta} J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) \left(r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) + \gamma \hat{V}_{\phi}^{\pi}(\mathbf{s}_{i,t+1}) - \hat{V}_{\phi}^{\pi}(\mathbf{s}_{i,t}) \right)$$

+ lower variance (due to critic)
- not unbiased (if the critic is not perfect)

Policy gradient:

$$\nabla_{\theta} J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) \left(\left(\sum_{t'=t}^T \gamma^{t'-t} r(\mathbf{s}_{i,t'}, \mathbf{a}_{i,t'}) \right) - b \right)$$

+ no bias
- higher variance (because single-sample estimate)

can we use \hat{V}_{ϕ}^{π} and still keep the estimator unbiased?

$$\nabla_{\theta} J(\theta) \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(\mathbf{a}_{i,t} | \mathbf{s}_{i,t}) \left(\left(\sum_{t'=t}^T \gamma^{t'-t} r(\mathbf{s}_{i,t'}, \mathbf{a}_{i,t'}) \right) - \hat{V}_{\phi}^{\pi}(\mathbf{s}_{i,t}) \right)$$

+ no bias
+ lower variance (baseline is closer to rewards)

Eligibility Traces and N-step Returns

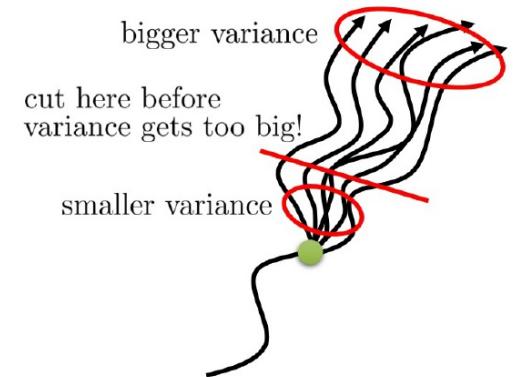
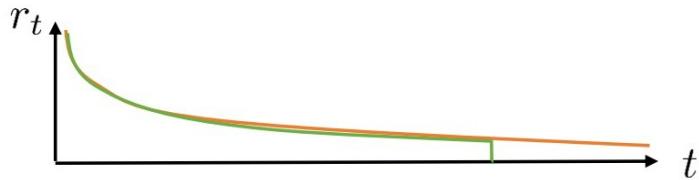
$$\hat{A}_C^\pi(s_t, a_t) = r(s_t, a_t) + \gamma \hat{V}_\phi^\pi(s_{t+1}) - \hat{V}_\phi^\pi(s_t)$$

- + lower variance
- higher bias if value is wrong (it always is)

$$\hat{A}_{MC}^\pi(s_t, a_t) = \sum_{t'=t}^{\infty} \gamma^{t'-t} r(s_{t'}, a_{t'}) - \hat{V}_\phi^\pi(s_t)$$

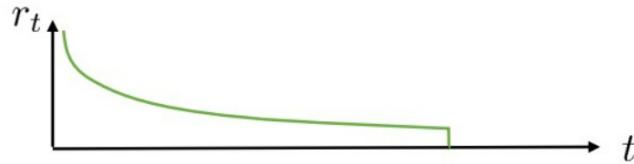
- + no bias
- higher variance (because single-sample estimate)

Can we combine these two, to control bias/variance tradeoff?



$$\hat{A}_n^\pi(s_t, a_t) = \sum_{t'=t}^{t+n} \gamma^{t'-t} r(s_{t'}, a_{t'}) - \hat{V}_\phi^\pi(s_t) + \gamma^n \hat{V}_\phi^\pi(s_{t+n})$$

Generalized Advantage Estimation



Do we have to choose just one n?

Cut everywhere all at once!

$$\hat{A}_n^\pi(\mathbf{s}_t, \mathbf{a}_t) = \sum_{t'=t}^{t+n} \gamma^{t'-t} r(\mathbf{s}_{t'}, \mathbf{a}_{t'}) - \hat{V}_\phi^\pi(\mathbf{s}_t) + \gamma^n \hat{V}_\phi^\pi(\mathbf{s}_{t+n})$$

$$\hat{A}_{\text{GAE}}^\pi(\mathbf{s}_t, \mathbf{a}_t) = \sum_{n=1}^{\infty} w_n \hat{A}_n^\pi(\mathbf{s}_t, \mathbf{a}_t)$$

weighted combination of n-step returns

How to weight?

Mostly prefer cutting earlier (less variance)

$$w_n \propto \lambda^{n-1}$$

exponential falloff

$$\hat{A}_{\text{GAE}}^\pi(\mathbf{s}_t, \mathbf{a}_t) = r(\mathbf{s}_t, \mathbf{a}_t) + \gamma((1-\lambda)\hat{V}_\phi^\pi(\mathbf{s}_{t+1}) + \lambda(r(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}) + \gamma((1-\lambda)\hat{V}_\phi^\pi(\mathbf{s}_{t+2}) + \lambda r(\mathbf{s}_{t+2}, \mathbf{a}_{t+2}) + \dots))$$

$$\hat{A}_{\text{GAE}}^\pi(\mathbf{s}_t, \mathbf{a}_t) = \sum_{t'=t}^{\infty} (\gamma \lambda)^{t'-t} \delta_{t'}$$

$$\delta_{t'} = r(\mathbf{s}_{t'}, \mathbf{a}_{t'}) + \gamma \hat{V}_\phi^\pi(\mathbf{s}_{t'+1}) - \hat{V}_\phi^\pi(\mathbf{s}_{t'})$$

most J_{t+1}, V_{t+1} & J_{t+2} ← Policy Grad v.s Q-Learning

off-policy + Approximation (TD) + Approximate function \rightarrow Value Function

Actor-Critic Algorithm: Batch vs. Online

batch actor-critic algorithm:

1. sample $\{s_i, a_i\}$ from $\pi_\theta(a|s)$ (run it on the robot)
2. fit $\hat{V}_\phi^\pi(s)$ to sampled reward sums
3. evaluate $\hat{A}^\pi(s_i, a_i) = r(s_i, a_i) + \gamma \hat{V}_\phi^\pi(s'_i) - \hat{V}_\phi^\pi(s_i)$
4. $\nabla_\theta J(\theta) \approx \sum_i \nabla_\theta \log \pi_\theta(a_i|s_i) \hat{A}^\pi(s_i, a_i)$
5. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

VS online actor-critic algorithm:

1. take action $a \sim \pi_\theta(a|s)$, get (s, a, s', r)
2. update \hat{V}_ϕ^π using target $r + \gamma \hat{V}_\phi^\pi(s')$
3. evaluate $\hat{A}^\pi(s, a) = r(s, a) + \gamma \hat{V}_\phi^\pi(s') - \hat{V}_\phi^\pi(s)$
4. $\nabla_\theta J(\theta) \approx \nabla_\theta \log \pi_\theta(a|s) \hat{A}^\pi(s, a)$
5. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

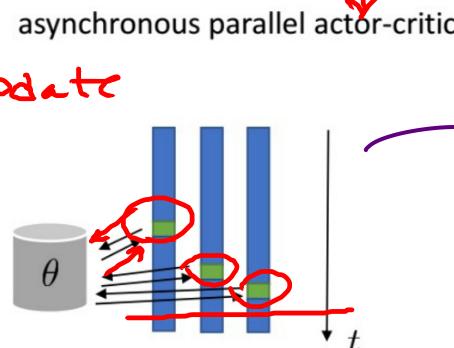
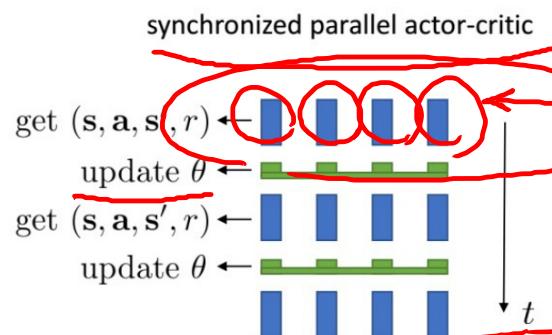
just 1 training sample

Online Actor-Critic in Practice: A2C and A3C

online actor-critic algorithm:

1. take action $\mathbf{a} \sim \pi_\theta(\mathbf{a}|\mathbf{s})$, get $(\mathbf{s}, \mathbf{a}, \mathbf{s}', r)$
2. update \hat{V}_ϕ^π using target $r + \gamma \hat{V}_\phi^\pi(\mathbf{s}') \leftarrow$
3. evaluate $\hat{A}^\pi(\mathbf{s}, \mathbf{a}) = r(\mathbf{s}, \mathbf{a}) + \gamma \hat{V}_\phi^\pi(\mathbf{s}') - \hat{V}_\phi^\pi(\mathbf{s})$
4. $\nabla_\theta J(\theta) \approx \nabla_\theta \log \pi_\theta(\mathbf{a}|\mathbf{s}) \hat{A}^\pi(\mathbf{s}, \mathbf{a}) \leftarrow$
5. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

works best with a batch (e.g., parallel workers)



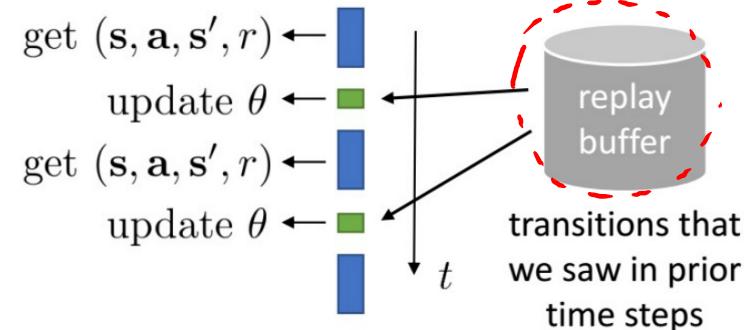
From On-Policy to Off-Policy Actor-Critic

online actor-critic algorithm:

1. take action $\mathbf{a} \sim \pi_\theta(\mathbf{a}|\mathbf{s})$, get $(\mathbf{s}, \mathbf{a}, \mathbf{s}', r)$
2. update \hat{V}_ϕ^π using target $r + \gamma \hat{V}_\phi^\pi(\mathbf{s}')$
3. evaluate $\hat{A}^\pi(\mathbf{s}, \mathbf{a}) = r(\mathbf{s}, \mathbf{a}) + \gamma \hat{V}_\phi^\pi(\mathbf{s}') - \hat{V}_\phi^\pi(\mathbf{s})$
4. $\nabla_\theta J(\theta) \approx \nabla_\theta \log \pi_\theta(\mathbf{a}|\mathbf{s}) \hat{A}^\pi(\mathbf{s}, \mathbf{a})$
5. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

Can we remove the on-policy assumption?

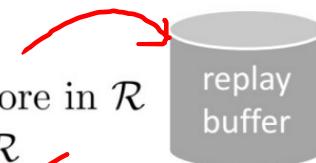
!! اپنے کمیں درست باشیں اون پلی سیتیکن
@@ اپنے کمیں درست باشیں اون پلی سیتیکن
off-policy actor-critic
 $\rightarrow (\mathbf{s}, \mathbf{a}, \mathbf{s}', r)$



From On-Policy to Off-Policy Actor-Critic

off-policy actor-critic algorithm:

1. take action $\mathbf{a} \sim \pi_\theta(\mathbf{a}|\mathbf{s})$, get $(\mathbf{s}, \mathbf{a}, \mathbf{s}', r)$, store in \mathcal{R}
2. sample a batch $\{\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}'_i\}$ from buffer \mathcal{R}
3. update \hat{V}_ϕ^π using targets $y_i = r_i + \gamma \hat{V}_\phi^\pi(\mathbf{s}'_i)$ for each \mathbf{s}_i
4. evaluate $\hat{A}^\pi(\mathbf{s}_i, \mathbf{a}_i) = r(\mathbf{s}_i, \mathbf{a}_i) + \gamma V_\phi^\pi(\mathbf{s}'_i) - \hat{V}_\phi^\pi(\mathbf{s}_i)$
5. $\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_i \nabla_\theta \log \pi_\theta(\mathbf{a}_i|\mathbf{s}_i) \hat{A}^\pi(\mathbf{s}_i, \mathbf{a}_i)$
6. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$



$$\mathcal{L}(\phi) = \frac{1}{N} \sum_i \left\| \hat{V}_\phi^\pi(\mathbf{s}_i) - y_i \right\|^2$$

not the right target value

not the action π_θ would have taken!

This algorithm is broken!

Can you spot the problems?

Off-Policy Actor-Critic: Fixing the Value Function

off-policy actor-critic algorithm:

1. take action $\mathbf{a} \sim \pi_\theta(\mathbf{a}|\mathbf{s})$, get $(\mathbf{s}, \mathbf{a}, \mathbf{s}', r)$, store in \mathcal{R}
2. sample a batch $\{\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}'_i\}$ from buffer \mathcal{R}
3. update \hat{V}_ϕ^π using targets $y_i = r_i + \gamma \hat{V}_\phi^\pi(\mathbf{s}'_i)$ for each \mathbf{s}_i
4. evaluate $\hat{A}^\pi(\mathbf{s}_i, \mathbf{a}_i) = r(\mathbf{s}_i, \mathbf{a}_i) + \gamma \hat{V}_\phi^\pi(\mathbf{s}'_i) - \hat{V}_\phi^\pi(\mathbf{s}_i)$
5. $\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_i \nabla_\theta \log \pi_\theta(\mathbf{a}_i|\mathbf{s}_i) \hat{A}^\pi(\mathbf{s}_i, \mathbf{a}_i)$
6. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

3. update \hat{Q}_ϕ^π using targets $y_i = r_i + \gamma \hat{V}_\phi^\pi(\mathbf{s}'_i)$ for each $\mathbf{s}_i, \mathbf{a}_i$

$$Q(\mathbf{s}_i, \mathbf{a}_i)$$

$$(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}', r)$$

$$(\mathbf{s}_t, \hat{\mathbf{a}}_t)$$

$$V^\pi(\mathbf{s}_t) = \sum_{t'=t}^T E_{\pi_\theta} [r(\mathbf{s}_{t'}, \mathbf{a}_{t'})|\mathbf{s}_t] = E_{\mathbf{a} \sim \pi(\mathbf{a}_t|\mathbf{s}_t)} [Q(\mathbf{s}_t, \mathbf{a}_t)]$$

$$\cancel{V^\pi(\mathbf{s}_t) = \sum_{t'=t}^T E_{\pi_\theta} [r(\mathbf{s}_{t'}, \mathbf{a}_{t'})|\mathbf{s}_t]}$$

$$Q^\pi(\mathbf{s}_t, \mathbf{a}_t) = \sum_{t'=t}^T E_{\pi_\theta} [r(\mathbf{s}_{t'}, \mathbf{a}_{t'})|\mathbf{s}_t, \mathbf{a}_t]$$

not the right target value

$$\mathcal{L}(\phi) = \frac{1}{N} \sum_i \left\| \hat{Q}_\phi^\pi(\mathbf{s}_i, \mathbf{a}_i) - y_i \right\|^2$$

not from replay buffer \mathcal{R} !

$a \rightarrow$ 3 تا، 0، 1، 2، 4، 5، 6، 7، 8، 9، 10، 11، 12، 13، 14، 15، 16، 17، 18، 19، 20، 21، 22، 23، 24، 25، 26، 27، 28، 29، 30، 31، 32، 33، 34، 35، 36، 37، 38، 39، 40، 41، 42، 43، 44، 45، 46، 47، 48، 49، 50، 51، 52، 53، 54، 55، 56، 57، 58، 59، 60، 61، 62، 63، 64، 65، 66، 67، 68، 69، 70، 71، 72، 73، 74، 75، 76، 77، 78، 79، 80، 81، 82، 83، 84، 85، 86، 87، 88، 89، 90، 91، 92، 93، 94، 95، 96، 97، 98، 99، 100، 101، 102، 103، 104، 105، 106، 107، 108، 109، 110، 111، 112، 113، 114، 115، 116، 117، 118، 119، 120، 121، 122، 123، 124، 125، 126، 127، 128، 129، 130، 131، 132، 133، 134، 135، 136، 137، 138، 139، 140، 141، 142، 143، 144، 145، 146، 147، 148، 149، 150، 151، 152، 153، 154، 155، 156، 157، 158، 159، 160، 161، 162، 163، 164، 165، 166، 167، 168، 169، 170، 171، 172، 173، 174، 175، 176، 177، 178، 179، 180، 181، 182، 183، 184، 185، 186، 187، 188، 189، 190، 191، 192، 193، 194، 195، 196، 197، 198، 199، 200، 201، 202، 203، 204، 205، 206، 207، 208، 209، 210، 211، 212، 213، 214، 215، 216، 217، 218، 219، 220، 221، 222، 223، 224، 225، 226، 227، 228، 229، 230، 231، 232، 233، 234، 235، 236، 237، 238، 239، 240، 241، 242، 243، 244، 245، 246، 247، 248، 249، 250، 251، 252، 253، 254، 255، 256، 257، 258، 259، 260، 261، 262، 263، 264، 265، 266، 267، 268، 269، 270، 271، 272، 273، 274، 275، 276، 277، 278، 279، 280، 281، 282، 283، 284، 285، 286، 287، 288، 289، 290، 291، 292، 293، 294، 295، 296، 297، 298، 299، 300، 301، 302، 303، 304، 305، 306، 307، 308، 309، 310، 311، 312، 313، 314، 315، 316، 317، 318، 319، 320، 321، 322، 323، 324، 325، 326، 327، 328، 329، 330، 331، 332، 333، 334، 335، 336، 337، 338، 339، 340، 341، 342، 343، 344، 345، 346، 347، 348، 349، 350، 351، 352، 353، 354، 355، 356، 357، 358، 359، 360، 361، 362، 363، 364، 365، 366، 367، 368، 369، 370، 371، 372، 373، 374، 375، 376، 377، 378، 379، 380، 381، 382، 383، 384، 385، 386، 387، 388، 389، 390، 391، 392، 393، 394، 395، 396، 397، 398، 399، 400، 401، 402، 403، 404، 405، 406، 407، 408، 409، 410، 411، 412، 413، 414، 415، 416، 417، 418، 419، 420، 421، 422، 423، 424، 425، 426، 427، 428، 429، 430، 431، 432، 433، 434، 435، 436، 437، 438، 439، 440، 441، 442، 443، 444، 445، 446، 447، 448، 449، 450، 451، 452، 453، 454، 455، 456، 457، 458، 459، 460، 461، 462، 463، 464، 465، 466، 467، 468، 469، 470، 471، 472، 473، 474، 475، 476، 477، 478، 479، 480، 481، 482، 483، 484، 485، 486، 487، 488، 489، 490، 491، 492، 493، 494، 495، 496، 497، 498، 499، 500، 501، 502، 503، 504، 505، 506، 507، 508، 509، 510، 511، 512، 513، 514، 515، 516، 517، 518، 519، 520، 521، 522، 523، 524، 525، 526، 527، 528، 529، 530، 531، 532، 533، 534، 535، 536، 537، 538، 539، 540، 541، 542، 543، 544، 545، 546، 547، 548، 549، 550، 551، 552، 553، 554، 555، 556، 557، 558، 559، 560، 561، 562، 563، 564، 565، 566، 567، 568، 569، 570، 571، 572، 573، 574، 575، 576، 577، 578، 579، 580، 581، 582، 583، 584، 585، 586، 587، 588، 589، 589، 590، 591، 592، 593، 594، 595، 596، 597، 598، 599، 600، 601، 602، 603، 604، 605، 606، 607، 608، 609، 610، 611، 612، 613، 614، 615، 616، 617، 618، 619، 620، 621، 622، 623، 624، 625، 626، 627، 628، 629، 630، 631، 632، 633، 634، 635، 636، 637، 638، 639، 640، 641، 642، 643، 644، 645، 646، 647، 648، 649، 650، 651، 652، 653، 654، 655، 656، 657، 658، 659، 660، 661، 662، 663، 664، 665، 666، 667، 668، 669، 669، 670، 671، 672، 673، 674، 675، 676، 677، 678، 679، 679، 680، 681، 682، 683، 684، 685، 686، 687، 688، 689، 689، 690، 691، 692، 693، 694، 695، 696، 697، 698، 699، 699، 700، 701، 702، 703، 704، 705، 706، 707، 708، 709، 709، 710، 711، 712، 713، 714، 715، 716، 717، 718، 719، 719، 720، 721، 722، 723، 724، 725، 726، 727، 728، 729، 729، 730، 731، 732، 733، 734، 735، 736، 737، 738، 739، 739، 740، 741، 742، 743، 744، 745، 746، 747، 748، 748، 749، 750، 751، 752، 753، 754، 755، 756، 757، 758، 759، 759، 760، 761، 762، 763، 764، 765، 766، 767، 768، 769، 769، 770، 771، 772، 773، 774، 775، 776، 777، 778، 778، 779، 779، 780، 781، 782، 783، 784، 785، 786، 787، 788، 788، 789، 789، 790، 791، 792، 793، 794، 795، 796، 797، 798، 798، 799، 799، 800، 801، 802، 803، 804، 805، 806، 807، 808، 809، 809، 810، 811، 812، 813، 814، 815، 816، 817، 818، 819، 819، 820، 821، 822، 823، 824، 825، 826، 827، 828، 829، 829، 830، 831، 832، 833، 834، 835، 836، 837، 838، 838، 839، 839، 840، 841، 842، 843، 844، 845، 846، 847، 848، 848، 849، 849، 850، 851، 852، 853، 854، 855، 856، 857، 858، 859، 859، 860، 861، 862، 863، 864، 865، 866، 867، 868، 869، 869، 870، 871، 872، 873، 874، 875، 876، 877، 878، 878، 879، 879، 880، 881، 882، 883، 884، 885، 886، 887، 888، 888، 889، 889، 890، 891، 892، 893، 894، 895، 896، 897، 897، 898، 898، 899، 899، 900، 901، 902، 903، 904، 905، 906، 907، 908، 909، 909، 910، 911، 912، 913، 914، 915، 916، 917، 918، 919، 919، 920، 921، 922، 923، 924، 925، 926، 927، 928، 929، 929، 930، 931، 932، 933، 934، 935، 936، 937، 938، 938، 939، 939، 940، 941، 942، 943، 944، 945، 946، 947، 948، 948، 949، 949، 950، 951، 952، 953، 954، 955، 956، 957، 958، 959، 959، 960، 961، 962، 963، 964، 965، 966، 967، 968، 969، 969، 970، 971، 972، 973، 974، 975، 976، 977، 978، 978، 979، 979، 980، 981، 982، 983، 984، 985، 986، 987، 987، 988، 988، 989، 989، 990، 991، 992، 993، 994، 995، 996، 997، 997، 998، 998، 999، 999، 1000، 1001، 1002، 1003، 1004، 1005، 1006، 1007، 1008، 1009، 1009، 1010، 1011، 1012، 1013، 1014، 1015، 1016، 1017، 1018، 1019، 1019، 1020، 1021، 1022، 1023، 1024، 1025، 1026، 1027، 1028، 1029، 1029، 1030، 1031، 1032، 1033، 1034، 1035، 1036، 1037، 1038، 1038، 1039، 1039، 1040، 1041، 1042، 1043، 1044، 1045، 1046، 1047، 1048، 1048، 1049، 1049، 1050، 1051، 1052، 1053، 1054، 1055، 1056، 1057، 1058، 1059، 1059، 1060، 1061، 1062، 1063، 1064، 1065، 1066، 1067، 1068، 1069، 1069، 1070، 1071، 1072، 1073، 1074، 1075، 1076، 1077، 1078، 1078، 1079، 1079، 1080، 1081، 1082، 1083، 1084، 1085، 1086، 1087، 1088، 1088، 1089، 1089، 1090، 1091، 1092، 1093، 1094، 1095، 1096، 1097، 1097، 1098، 1098، 1099، 1099، 1100، 1101، 1102، 1103، 1104، 1105، 1106، 1107، 1108، 1109، 1109، 1110، 1111، 1112، 1113، 1114، 1115، 1116، 1117، 1118، 1119، 1119، 1120، 1121، 1122، 1123، 1124، 1125، 1126، 1127، 1128، 1129، 1129، 1130، 1131، 1132، 1133، 1134، 1135، 1136، 1137، 1138، 1138، 1139، 1139، 1140، 1141، 1142، 1143، 1144، 1145، 1146، 1147، 1148، 1148، 1149، 1149، 1150، 1151، 1152، 1153، 1154، 1155، 1156، 1157، 1158، 1159، 1159، 1160، 1161، 1162، 1163، 1164، 1165، 1166، 1167، 1168، 1169، 1169، 1170، 1171، 1172، 1173، 1174، 1175، 1176، 1177، 1178، 1178، 1179، 1179، 1180، 1181، 1182، 1183، 1184، 1185، 1186، 1187، 1188، 1188، 1189، 1189، 1190، 1191، 1192، 1193، 1194، 1195، 1196، 1197، 1197، 1198، 1198، 1199، 1199، 1200، 1201، 1202، 1203، 1204، 1205، 1206، 1207، 1208، 1209، 1209، 1210، 1211، 1212، 1213، 1214، 1215، 1216، 1217، 1218، 1219، 1219، 1220، 1221، 1222، 1223، 1224، 1225، 1226، 1227، 1228، 1229، 1229، 1230، 1231، 1232، 1233، 1234، 1235، 1236، 1237، 1238، 1238، 1239، 1239، 1240، 1241، 1242، 1243، 1244، 1245، 1246، 1247، 1248، 1248، 1249، 1249، 1250، 1251، 1252، 1253، 1254، 1255، 1256، 1257، 1258، 1259، 1259، 1260، 1261، 1262، 1263، 1264، 1265، 1266، 1267، 1268، 1269، 1269، 1270، 1271، 1272، 1273، 1274، 1275، 1276، 1277، 1278، 1278، 1279، 1279، 1280، 1281، 1282، 1283، 1284، 1285، 1286، 1287، 1288، 1288، 1289، 1289، 1290، 1291، 1292، 1293، 1294، 1295، 1296، 1297، 1297، 1298، 1298، 1299، 1299، 1300، 1301، 1302، 1303، 1304، 1305، 1306، 1307، 1308، 1309، 1309، 1310، 1311، 1312، 1313، 1314، 1315، 1316، 1317، 1318، 1319، 1319، 1320، 1321، 1322، 1323، 1324، 1325، 1326، 1327، 1328، 1329، 1329، 1330، 1331، 1332، 1333، 1334، 1335، 1336، 1337، 1338، 1338، 1339، 1339، 1340، 1341، 1342، 1343، 1344، 1345، 1346، 1347، 1348، 1348، 1349، 1349، 1350، 1351، 1352، 1353، 1354، 1355، 1356، 1357، 1358، 1359، 1359، 1360، 1361، 1362، 1363، 1364، 1365، 1366، 1367، 1368، 1369، 1369، 1370، 1371، 1372، 1373، 1374، 1375، 1376، 1377، 1378، 1378، 1379، 1379، 1380، 1381، 1382، 1383، 1384، 1385، 1386، 1387، 1388، 1388، 1389، 1389، 1390، 1391، 1392، 1393، 1394، 1395، 1396، 1397، 1397، 1398، 1398، 1399، 1399، 1400، 1401، 1402، 1403، 1404، 1405، 1406، 1407، 1408، 1409، 1409، 1410، 1411، 1412، 1413، 1414، 1415، 1416، 1417، 1418، 1419، 1419، 1420، 1421، 1422، 1423، 1424، 1425، 1426، 1427، 1428، 1429، 1429، 1430، 1431، 1432، 1433، 1434، 1435، 1436، 1437، 1438، 1438، 1439، 1439، 1440، 1441، 1442، 1443، 1444، 1445، 1446، 1447، 1448، 1448، 1449، 1449، 1450، 1451، 1452، 1453، 1454، 1455، 1456، 1457، 1458، 1459، 1459، 1460، 1461، 1462، 1463، 1464، 1465، 1466، 1467، 1468، 1469، 1469، 1470، 1471، 1472، 1473، 1474، 1475، 1476، 1477، 1478، 1478، 1479، 1479، 1480، 1481، 1482، 1483، 1484، 1485، 1486، 1487، 1488، 1488، 1489، 1489، 1490، 1491، 1492، 1493، 1494، 1495، 1496، 1497، 1497، 1498، 1498، 1499، 1499، 1500، 1501، 1502، 1503، 1504، 1505، 1506، 1507، 1508، 1509، 1509، 1510، 1511، 1512، 1513، 1514، 1515، 1516، 1517، 1518، 1519، 1519، 1520، 1521، 1522، 1523، 1524، 1525، 1526، 1527، 1528، 1529، 1529، 1530، 1531، 1532، 1533، 1534، 1535، 1536، 1537، 1538، 1538، 1539، 1539، 1540، 1541، 1542، 1543، 1544، 1545، 1546، 1547، 1548، 1548، 1549، 1549، 1550، 1551، 1552، 1553، 1554، 1555، 1556، 1557، 1558، 1559، 1559، 1560، 1561، 1562، 1563، 1564، 1565، 1566، 1567، 1568، 1569، 1569، 1570، 1571، 1572، 1573، 1574، 1575، 1576، 1577، 1578، 1578، 1579، 1579، 1580، 1581، 1582، 1583، 1584، 1585، 1586، 1587، 1588، 1588، 1589، 1589، 1590، 1591، 1592، 1593، 1594، 1595، 1596، 1597، 1597، 1598، 1598، 1599، 1599، 1600، 1601، 1602، 1603، 1604، 1605، 1606، 1607، 1608، 1609، 1609، 1610، 1611، 1612، 1613، 1614، 1615، 1616، 1617، 1618، 1619، 1619، 1620، 1621، 1622، 1623، 1624، 1625، 1626، 1627، 1628، 1629، 1629، 1630، 1631، 1632، 1633، 1634، 1635، 1636، 1637، 1638، 1638، 1639، 1639، 1640، 1641، 1642، 1643، 1644، 1645، 1646، 1647، 1648، 1648، 1649، 1649، 1650، 1651، 1652، 1653، 1654، 1655، 1656، 1657، 1658، 1659، 1659، 1660، 1661، 1662، 1663، 1664، 1665، 1666، 1667، 1668، 1669، 1669، 1670، 1671، 1672، 1673، 1674، 1675، 1676، 1677، 1678، 1678، 1679، 1679، 1680، 1681، 1682، 1683، 1684، 1685، 1686، 1687، 1688، 1688، 1689، 1689، 1690، 1691، 1692، 1693، 1694، 1695، 1696، 1697، 1697، 1698، 1698، 1699، 1699، 1700،

Off-Policy Actor-Critic: Fixing the Policy Update

off-policy actor-critic algorithm:

1. take action $\mathbf{a} \sim \pi_\theta(\mathbf{a}|\mathbf{s})$, get $(\mathbf{s}, \mathbf{a}, \mathbf{s}', r)$, store in \mathcal{R}
2. sample a batch $\{\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}'_i\}$ from buffer \mathcal{R}
3. update \hat{Q}_ϕ^π using targets $y_i = r_i + \gamma \hat{Q}_\phi^\pi(\mathbf{s}'_i, \mathbf{a}'_i)$ for each $\mathbf{s}_i, \mathbf{a}_i$
4. evaluate $\hat{A}^\pi(\mathbf{s}_i, \mathbf{a}_i) = Q(\mathbf{s}_i, \mathbf{a}_i) - \hat{V}_\phi^\pi(\mathbf{s}_i)$
5. $\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_i \nabla_\theta \log \pi_\theta(\mathbf{a}_i|\mathbf{s}_i) \hat{A}^\pi(\mathbf{s}_i, \mathbf{a}_i)$
6. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

$$\int \mathcal{P}(s) \mathcal{P}(a) \Delta \theta$$

not the action π_θ would have taken!

use the same trick, but this time for \mathbf{a}_i rather than \mathbf{a}'_i !

sample $\mathbf{a}_i^\pi \sim \pi_\theta(\mathbf{a}|\mathbf{s}_i)$ $\rightarrow \nabla_\theta J(\theta) \approx \frac{1}{N} \sum_i \nabla_\theta \log \pi_\theta(\mathbf{a}_i^\pi|\mathbf{s}_i) \hat{A}^\pi(\mathbf{s}_i, \mathbf{a}_i^\pi)$

in practice: $\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_i \nabla_\theta \log \pi_\theta(\mathbf{a}_i^\pi|\mathbf{s}_i) \hat{Q}^\pi(\mathbf{s}_i, \mathbf{a}_i^\pi)$

higher variance, but convenient
why is higher variance OK here?

1.
C.
.

Off-Policy Actor-Critic

off-policy actor-critic algorithm:

1. take action $\mathbf{a} \sim \pi_\theta(\mathbf{a}|\mathbf{s})$, get $(\mathbf{s}, \mathbf{a}, \mathbf{s}', r)$, store in \mathcal{R}
2. sample a batch $\{\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}'_i\}$ from buffer \mathcal{R}
3. update \hat{Q}_ϕ^π using targets $y_i = r_i + \gamma \hat{Q}_\phi^\pi(\mathbf{s}'_i, \mathbf{a}'_i)$ for each $\mathbf{s}_i, \mathbf{a}_i$
4. $\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_i \nabla_\theta \log \pi_\theta(\mathbf{a}_i^\pi | \mathbf{s}_i) Q^\pi(\mathbf{s}_i, \mathbf{a}_i^\pi)$ where $\mathbf{a}_i^\pi \sim \pi_\theta(\mathbf{a} | \mathbf{s}_i)$
5. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

Is there any remaining problem?

\mathbf{s}_i didn't come from $p_\theta(\mathbf{s})$

nothing we can do here, just accept it

intuition: we want optimal policy on $p_\theta(\mathbf{s})$
but we get optimal policy on a *broader* distribution