PPO and GAE

Annoucements

Will release the next reading Quiz on the PPO technical report

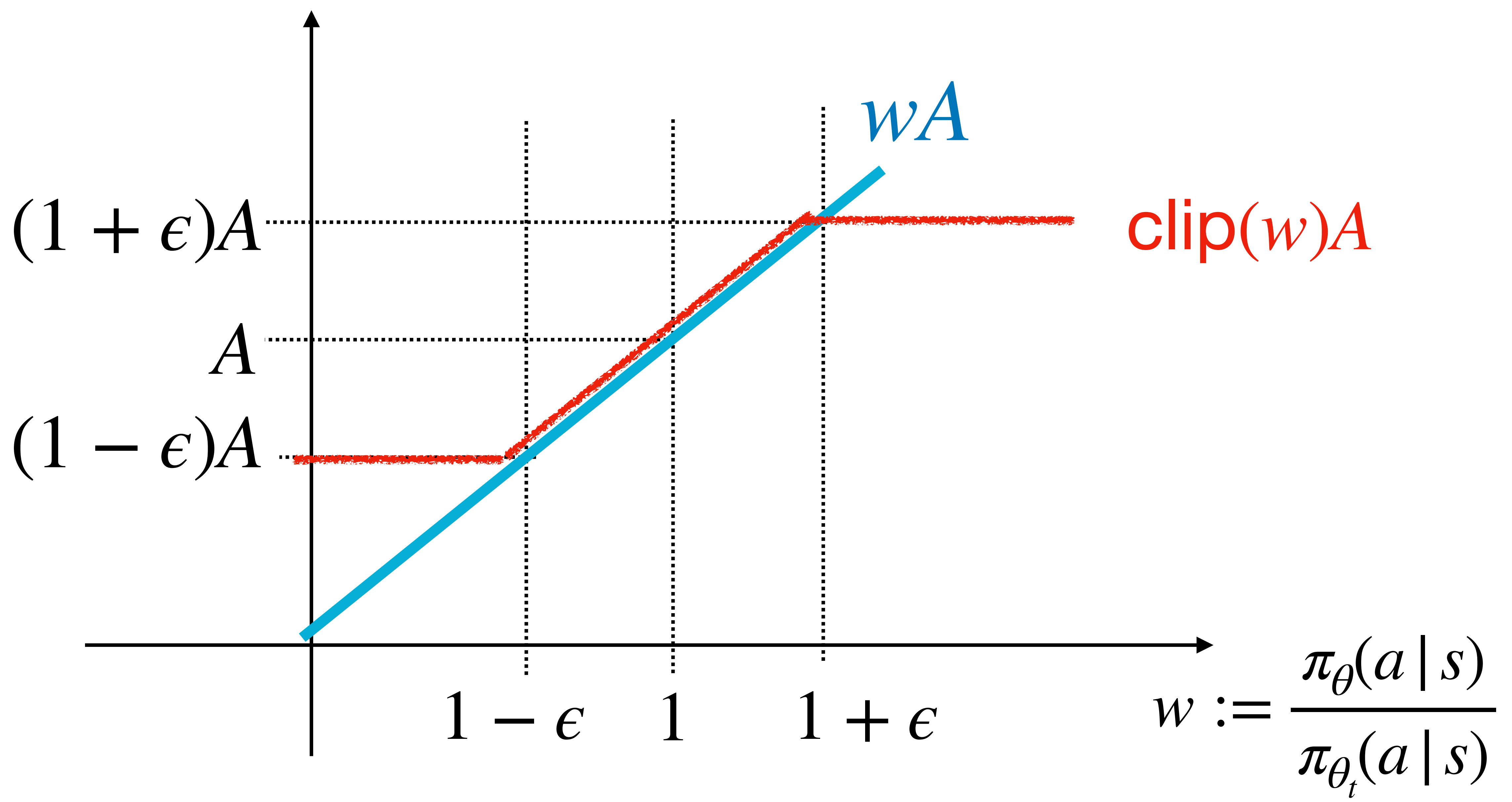
Will release the next programming assignment on NPG and PPO

Recap: Proximal Policy Optimization (PPO)

Policy optimization objective:

$$\hat{\ell}_{final}(\theta) = \sum_{s,a} \min \left\{ \frac{\pi_\theta(a|s)}{\pi_{\theta_t}(a|s)} \cdot A^{\pi_{\theta_t}}(s,a), \quad \text{clip}\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_t}(a|s)}, 1-\epsilon, 1+\epsilon \right) \cdot A^{\pi_{\theta_t}}(s,a) \right\}$$

When $A^{\pi_{\theta_t}}(s,a) > 0$, we want to increase the ratio $\pi_\theta(a|s)/\pi_{\theta_t}(a|s)$



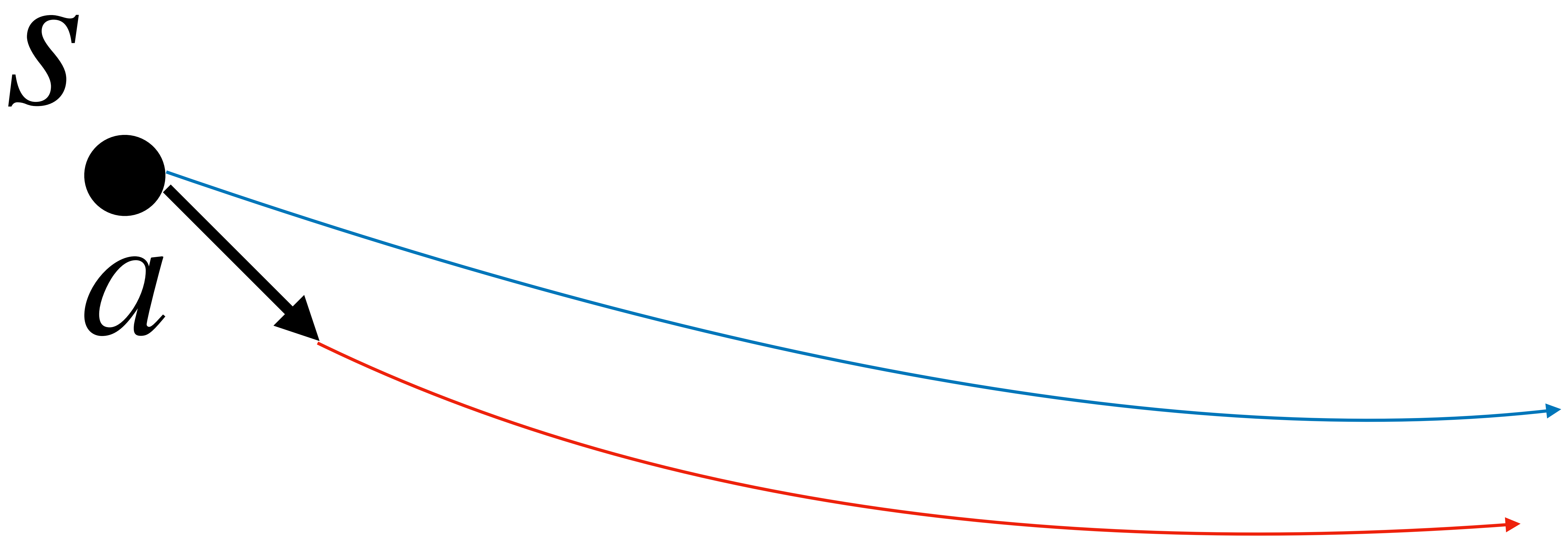
Main Question Today:

$$\hat{\ell}_{final}(\theta) = \sum_{s,a} \min \left\{ \frac{\pi_\theta(a \mid s)}{\pi_{\theta_t}(a \mid s)} \cdot A^{\pi_{\theta_t}}(s,a), \quad \text{clip}\left(\frac{\pi_\theta(a \mid s)}{\pi_{\theta_t}(a \mid s)}, 1-\epsilon, 1+\epsilon \right) \cdot A^{\pi_{\theta_t}}(s,a) \right\}$$

How to get these advantage values?

Attempt 1: Monte Carlo (MC) method

Given (s, a), estimate $A^\pi(s, a)$ via two MC rollouts



Can have high variance;

Require the ability to **reset**: i.e., go back to s again, and do one more rollout

Attempt 1: Monte Carlo (MC) method

Q: given (s, a), can we estimate $A^\pi(s, a)$ via one rollout?

$$A^\pi(s, a)/2 = (Q^\pi(s, a) - V^\pi(s))/2 = \mathbb{E}_{z \sim \text{Uniform}(\{0,1\})}(zQ^\pi(s, a) - (1 - z)V^\pi)$$

Can have high variance still

To further reduce variance, we will give up unbiaseness,
and trade bias for variance

Generalized Advantage Estimation

Our goal: estimate $A^{\pi}(s, a)$

We will do the following two steps:

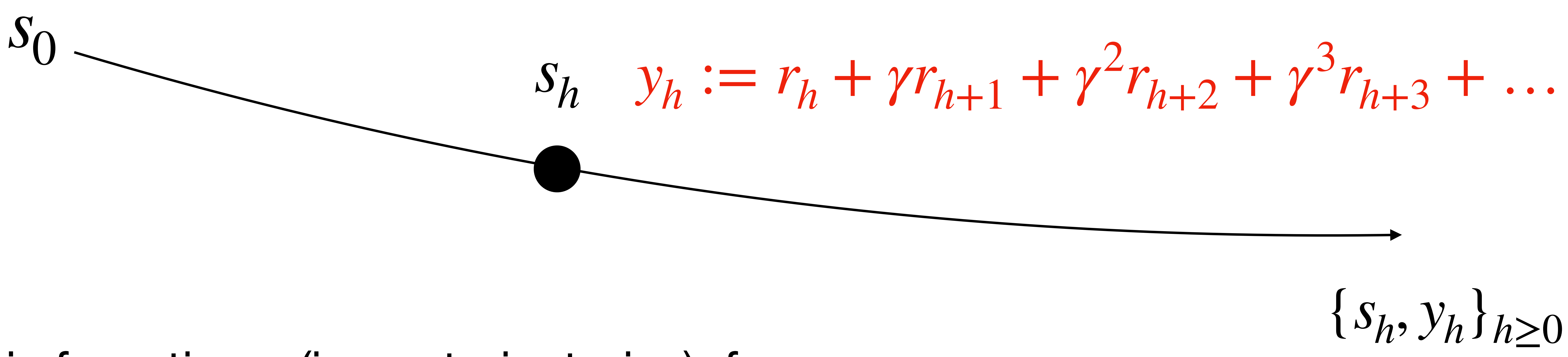
1. Estimate $V^{\pi}(s)$ using function approximation (neural network, decision tree, etc)

2. Form an estimate of $A^{\pi}(s, a)$ using the value function estimator V

Generalized Advantage Estimation

Estimate V^π

1. sample a trajectory $\tau \sim \pi$, for all h:



$$y_h := r_h + \gamma r_{h+1} + \gamma^2 r_{h+2} + \gamma^3 r_{h+3} + \ldots$$

2. Repeat this for n times (i.e., n trajectories), form a regression dataset:

$$\mathcal{D}_\pi = \{s, y\}$$

Generalized Advantage Estimation

Estimate V^{π}

Given $\mathscr{D}_{\pi} = \{s, y\}$, perform regression

$$\hat{V}^{\pi} = \arg\min_{V} \sum_{s,y \in \mathscr{D}_{\pi}} \left(V(s) - y\right)^2$$

The Bayes optimal for this regression is $\mathbb{E}[y \mid s] = V^{\pi}(s)$,
so if regression works well we can have $\hat{V}^{\pi} \approx V^{\pi}$

Generalized Advantage Estimation

Our goal: estimate $A^{\pi}(s, a)$

We will do the following two steps:

1. Estimate $V^{\pi}(s)$ using function approximation (neural network, decision tree, etc)

2. Form an estimate of $A^{\pi}(s, a)$ using the value function estimator V

Generalized Advantage Estimation

Denote V as an estimator of V^π, let's compute estimate for $A^\pi(s, a)$

$$\hat{A}^{(1)}(s_h, a_h) = r_h + \gamma V(s_{h+1}) - V(s_h)$$

(Low variance but can be highly biased)

$$\hat{A}^{(2)}(s_h, a_h) = \underbrace{r_h + \gamma r_{h+1} + \gamma^2 V(s_{h+2})}_{\approx Q^\pi(s_h, a_h)} - V(s_h)$$

(Slighly higher var but lower bias)

$$\hat{A}^{(3)}(s_h, a_h) = \underbrace{r_h + \gamma r_{h+1} + \gamma^2 r_{h+2} + \gamma^3 V(s_{h+2})}_{\approx Q^\pi(s_h, a_h)} - V(s_h)$$

(higher var but lower bias)

Q: What is $\hat{A}^\infty(s_h, a_h)$? Would using $\hat{A}^\infty(s_h, a_h)$ in the policy gradient have any bias issue?

Generalized Advantage Estimation

GAE uses an **exponential average** to combine these advantage estimators together

$$\lambda \in (0,1)$$

$$\hat{A}^{gae} = (1-\lambda)\left(\hat{A}^{(1)} + \lambda\hat{A}^{(2)} + \lambda^2\hat{A}^{(3)} + \ldots\ldots\right)$$

Generalized Advantage Estimation

Expressing $\hat{A}^{gae}$ using Bellman errors (Bellman residuals): $BE_\tau = r_\tau + \gamma V(s_{\tau+1}) - V(s_\tau)$

$$\hat{A}^{(1)}(s_h, a_h) = r_h + \gamma V(s_{h+1}) - V(s_h) = BE_h$$

$$\hat{A}^{(2)}(s_h, a_h) = r_h + \gamma r_{h+1} + \gamma^2 V(s_{h+2}) - V(s_h) = BE_h + \gamma BE_{h+1}$$

$$\hat{A}^{(3)}(s_h, a_h) = r_h + \gamma r_{h+1} + \gamma^2 r_{h+2} + \gamma^3 V(s_{h+2}) - V(s_h) = BE_h + \gamma BE_{h+1} + \gamma^2 BE_{h+2}$$

$$\hat{A}^{(k)}(s_h, a_h) = BE_h + \gamma BE_{h+1} + \gamma^2 BE_{h+2} + \ldots \gamma^{k-1} BE_{h+k-1}$$

$$(1-\lambda)(A^{(1)} + \lambda A^{(2)} + \lambda^2 A^{(3)} + \ldots) = \sum_{l=0}^{\infty} (\gamma\lambda)^l BE_{h+l}$$

Generalized Advantage Estimation

Expressing $\hat{A}^{gae}$ using Bellman errors (Bellman residuals): $BE_\tau = r_\tau + \gamma V(s_{\tau+1}) - V(s_\tau)$

$$(1-\lambda)(A^{(1)} + \lambda A^{(2)} + \lambda^2 A^{(3)} + \ldots) = \sum_{l=0}^{\infty} (\gamma\lambda)^l BE_{h+l}$$

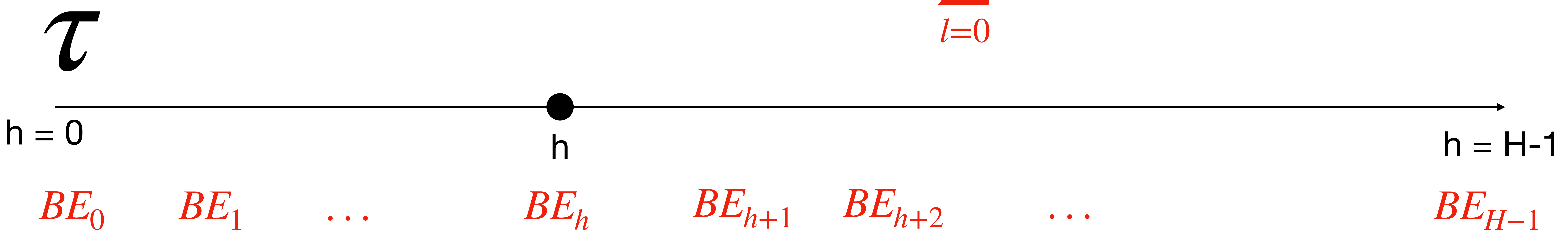
When $\lambda = 0$, the GAE estimate becomes

When $\lambda = 1$, the GAE estimate becomes

Generalized Advantage Estimation

In summary, given a trajectory $\tau \sim \pi$ of length H, and $V \approx V^{\pi}$, GAE copmutes $\hat{A}^{gae}(s_h, a_h)$ for all s_h, a_h on τ

$$(1-\lambda)(A_h^{(1)} + \lambda A_h^{(2)} + \lambda^2 A_h^{(3)} + \ldots) = \sum_{l=0}^{H-1} (\gamma\lambda)^l BE_{h+l}$$



$$\forall h: A_h^{gae} = BE_h + (\gamma\lambda) BE_{h+1} + (\gamma\lambda)^2 BE_{h+2} \ldots \ldots (\gamma\lambda)^{H-h-1} BE_{H-1}$$

Q: can you think about how to compute A_h^{gae} recursively using A_{h+1}^{gae} in a backward fashion?

Put everything together: PPO w/ GAE:

Initialize π_{θ_0} for the policy and V_{ω_0} for value function

For $t = 0 \to T$:

Run π_{θ_t} to collect multiple trajectories $\tau^1, \ldots, \tau^n$

Form the regression dataset $\mathcal{D} = \{s, y\}$ using the trajectories

Form A^{gae}: for each (s, a) in each trajectory, compute $A^{gae}(s, a)$ using V_{ω_t}

Construct policy loss: $\ell_\pi(\theta) = -\sum_{s,a} \min \left\{ \frac{\pi_\theta(a \mid s)}{\pi_{\theta_t}(a \mid s)} \cdot A^{gae}(s, a), \quad \text{clip}\left(\frac{\pi_\theta(a \mid s)}{\pi_{\theta_t}(a \mid s)}, 1 - \epsilon, 1 + \epsilon \right) \cdot A^{gae}(s, a) \right\}$

Construct V loss: $\ell_V(\omega) = \sum_{s,y} (V_\omega(s) - y)^2$

Update π and V via a few gradient updates on the combined loss $\ell_\pi(\theta) + \ell_V(\omega)$

Summary

PPO can be complicated, e.g., think about how many hyperparameters are there already?

There are further tricks to reduce variance, see the handout of the next programming assignment

Need to get your hands dirty and try it out in practice!