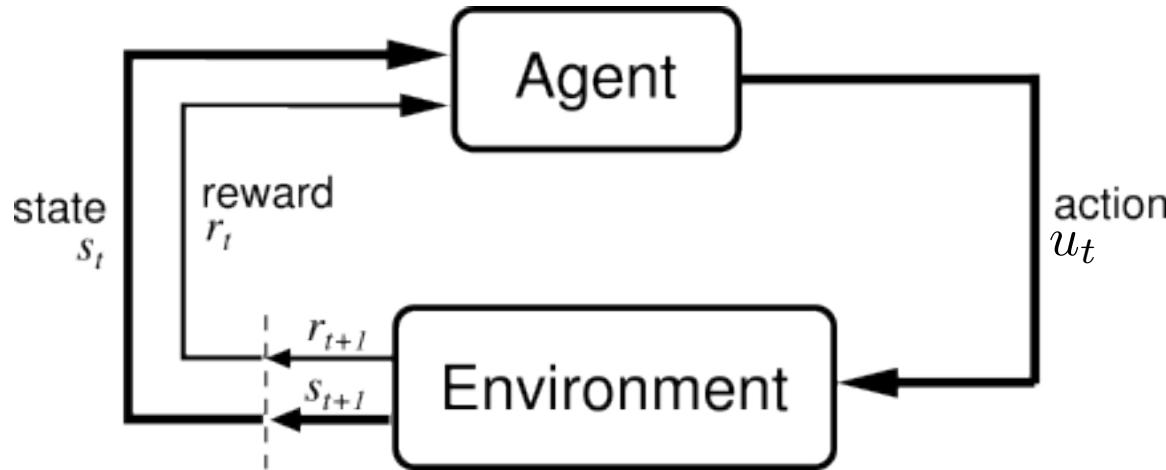


Policy Gradients

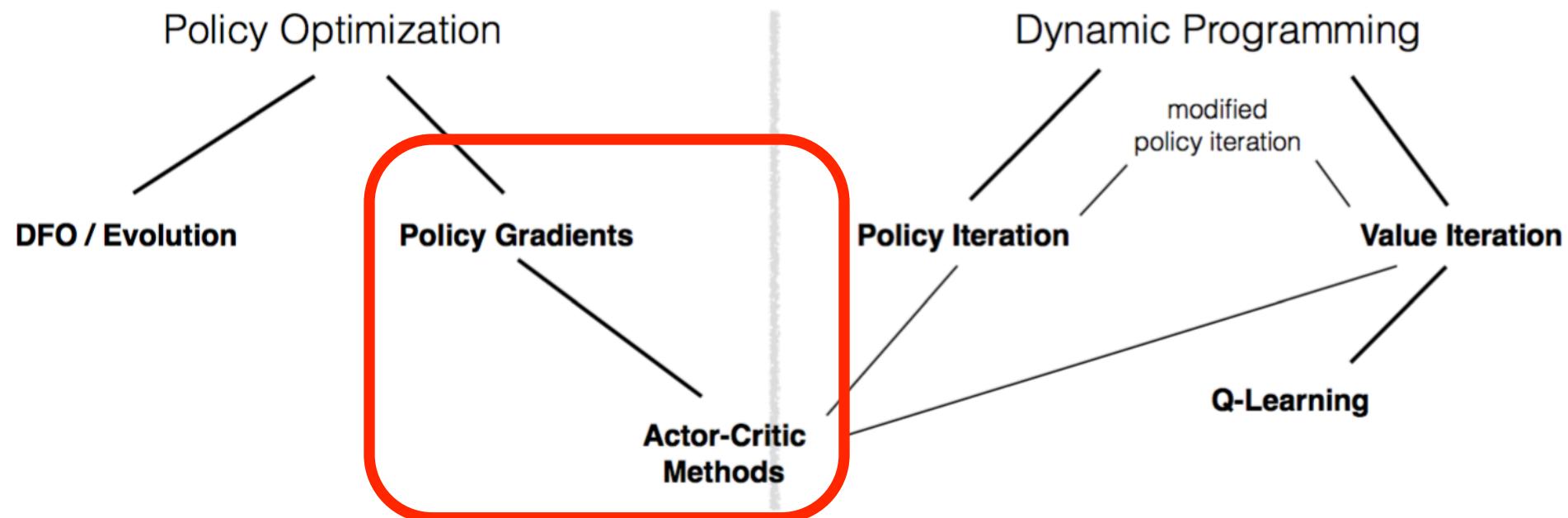
Pieter Abbeel – OpenAI / UC Berkeley / Gradescope

Slides authored with John Schulman and Xi (Peter) Chen

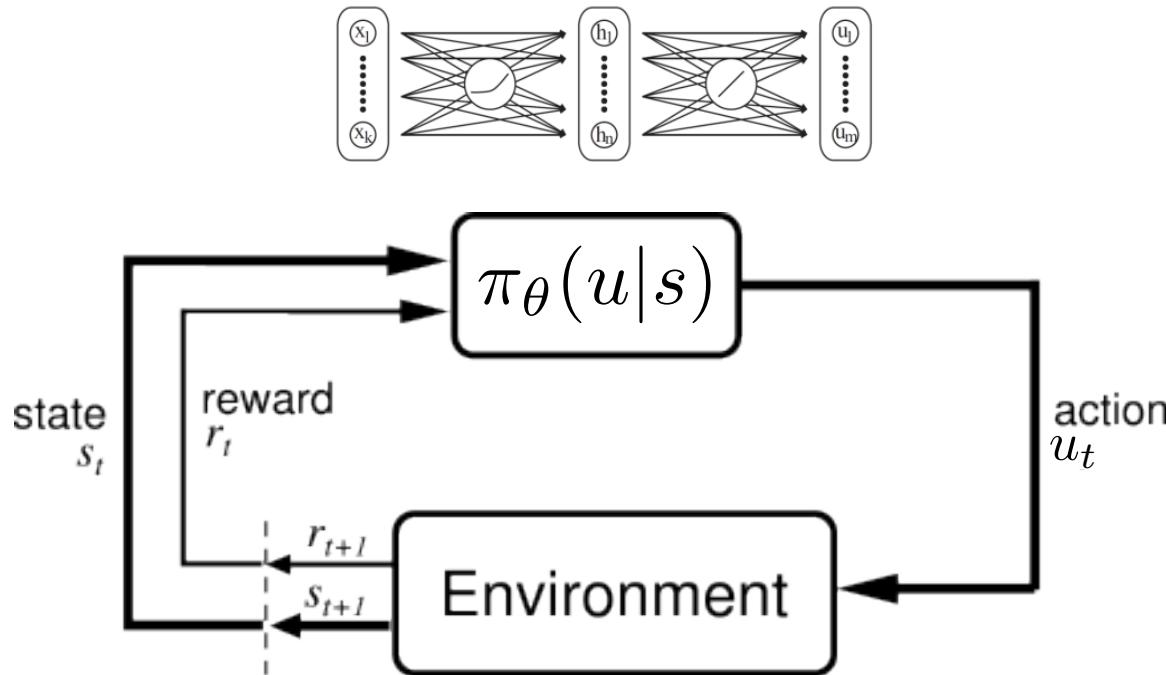
Reinforcement Learning



Policy Optimization in the RL Landscape



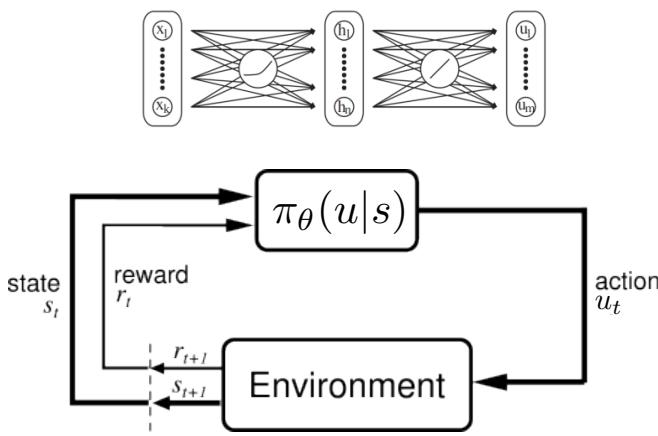
Policy Optimization



Policy Optimization

- Consider control policy parameterized by parameter vector θ

$$\max_{\theta} \mathbb{E}\left[\sum_{t=0}^H R(s_t) | \pi_{\theta}\right]$$



- Stochastic policy class (smooths out the problem):

$\pi_{\theta}(u|s)$: probability of action u in state s

Why Policy Optimization

- Often π can be simpler than Q or V
 - E.g., robotic grasp
- V: doesn't prescribe actions
 - Would need dynamics model (+ compute 1 Bellman back-up)
- Q: need to be able to efficiently solve $\arg \max_u Q_\theta(s, u)$
 - Challenge for continuous / high-dimensional action spaces*

*some recent work (partially) addressing this:

NAF: Gu, Lillicrap, Sutskever, Levine ICML 2016

Input Convex NNs: Amos, Xu, Kolter arXiv 2016

Deep Energy Q: Haarnoja, Tang, Abbeel, Levine, ICML 2017

Policy Optimization

- Conceptually:
- Empirically:

Optimize what you care about

More compatible with rich architectures (including recurrence)

More versatile

More compatible with auxiliary objectives

Dynamic Programming

Indirect, exploit the problem structure, self-consistency

More compatible with exploration and off-policy learning

More sample-efficient when they work

Likelihood Ratio Policy Gradient

We let τ denote a state-action sequence $s_0, u_0, \dots, s_H, u_H$. We overload notation: $R(\tau) = \sum_{t=0}^H R(s_t, u_t)$.

$$U(\theta) = \mathbb{E}\left[\sum_{t=0}^H R(s_t, u_t); \pi_\theta\right] = \sum_{\tau} P(\tau; \theta)R(\tau)$$

In our new notation, our goal is to find θ :

$$\max_{\theta} U(\theta) = \max_{\theta} \sum_{\tau} P(\tau; \theta)R(\tau)$$

Likelihood Ratio Policy Gradient

$$U(\theta) = \sum_{\tau} P(\tau; \theta) R(\tau)$$

Taking the gradient w.r.t. θ gives

$$\nabla_{\theta} U(\theta) = \nabla_{\theta} \sum_{\tau} P(\tau; \theta) R(\tau)$$

[Aleksandrov, Sysoyev, & Shemeneva, 1968]

[Rubinstein, 1969]

[Glynn, 1986]

[Reinforce, Williams 1992]

[GPOMDP, Baxter & Bartlett, 2001]

Likelihood Ratio Policy Gradient

$$U(\theta) = \sum_{\tau} P(\tau; \theta) R(\tau)$$

Taking the gradient w.r.t. θ gives

$$\begin{aligned}\nabla_{\theta} U(\theta) &= \nabla_{\theta} \sum_{\tau} P(\tau; \theta) R(\tau) \\ &= \sum_{\tau} \nabla_{\theta} P(\tau; \theta) R(\tau)\end{aligned}$$

[Aleksandrov, Sysoyev, & Shemeneva, 1968]

[Rubinstein, 1969]

[Glynn, 1986]

[Reinforce, Williams 1992]

[GPOMDP, Baxter & Bartlett, 2001]

Likelihood Ratio Policy Gradient

$$U(\theta) = \sum_{\tau} P(\tau; \theta) R(\tau)$$

Taking the gradient w.r.t. θ gives

$$\begin{aligned}\nabla_{\theta} U(\theta) &= \nabla_{\theta} \sum_{\tau} P(\tau; \theta) R(\tau) \\ &= \sum_{\tau} \nabla_{\theta} P(\tau; \theta) R(\tau) \\ &= \sum_{\tau} \frac{P(\tau; \theta)}{P(\tau; \theta)} \nabla_{\theta} P(\tau; \theta) R(\tau)\end{aligned}$$

[Aleksandrov, Sysoyev, & Shemeneva, 1968]

[Rubinstein, 1969]

[Glynn, 1986]

[Reinforce, Williams 1992]

[GPOMDP, Baxter & Bartlett, 2001]

Likelihood Ratio Policy Gradient

$$U(\theta) = \sum_{\tau} P(\tau; \theta) R(\tau)$$

Taking the gradient w.r.t. θ gives

$$\begin{aligned}\nabla_{\theta} U(\theta) &= \nabla_{\theta} \sum_{\tau} P(\tau; \theta) R(\tau) \\ &= \sum_{\tau} \nabla_{\theta} P(\tau; \theta) R(\tau) \\ &= \sum_{\tau} \frac{P(\tau; \theta)}{P(\tau; \theta)} \nabla_{\theta} P(\tau; \theta) R(\tau) \\ &= \sum_{\tau} P(\tau; \theta) \frac{\nabla_{\theta} P(\tau; \theta)}{P(\tau; \theta)} R(\tau)\end{aligned}$$

[Aleksandrov, Sysoyev, & Shemeneva, 1968]

[Rubinstein, 1969]

[Glynn, 1986]

[Reinforce, Williams 1992]

[GPOMDP, Baxter & Bartlett, 2001]

Likelihood Ratio Policy Gradient

$$U(\theta) = \sum_{\tau} P(\tau; \theta) R(\tau)$$

Taking the gradient w.r.t. θ gives

$$\begin{aligned}\nabla_{\theta} U(\theta) &= \nabla_{\theta} \sum_{\tau} P(\tau; \theta) R(\tau) \\ &= \sum_{\tau} \nabla_{\theta} P(\tau; \theta) R(\tau) \\ &= \sum_{\tau} \frac{P(\tau; \theta)}{P(\tau; \theta)} \nabla_{\theta} P(\tau; \theta) R(\tau) \\ &= \sum_{\tau} P(\tau; \theta) \frac{\nabla_{\theta} P(\tau; \theta)}{P(\tau; \theta)} R(\tau) \\ &= \sum_{\tau} P(\tau; \theta) \nabla_{\theta} \log P(\tau; \theta) R(\tau)\end{aligned}$$

[Aleksandrov, Sysoyev, & Shemeneva, 1968]

[Rubinstein, 1969]

[Glynn, 1986]

[Reinforce, Williams 1992]

[GPOMDP, Baxter & Bartlett, 2001]

Likelihood Ratio Policy Gradient

$$U(\theta) = \sum_{\tau} P(\tau; \theta) R(\tau)$$

Taking the gradient w.r.t. θ gives

$$\begin{aligned}\nabla_{\theta} U(\theta) &= \nabla_{\theta} \sum_{\tau} P(\tau; \theta) R(\tau) \\ &= \sum_{\tau} \nabla_{\theta} P(\tau; \theta) R(\tau) \\ &= \sum_{\tau} \frac{P(\tau; \theta)}{P(\tau; \theta)} \nabla_{\theta} P(\tau; \theta) R(\tau) \\ &= \sum_{\tau} P(\tau; \theta) \frac{\nabla_{\theta} P(\tau; \theta)}{P(\tau; \theta)} R(\tau) \\ &= \sum_{\tau} P(\tau; \theta) \nabla_{\theta} \log P(\tau; \theta) R(\tau)\end{aligned}$$

Approximate with the empirical estimate for m sample paths under policy π_{θ} :

$$\nabla_{\theta} U(\theta) \approx \hat{g} = \frac{1}{m} \sum_{i=1}^m \nabla_{\theta} \log P(\tau^{(i)}; \theta) R(\tau^{(i)})$$

[Aleksandrov, Sysoyev, & Shemeneva, 1968]

[Rubinstein, 1969]

[Glynn, 1986]

[Reinforce, Williams 1992]

[GPOMDP, Baxter & Bartlett, 2001]

Likelihood Ratio Gradient: Validity

$$\nabla U(\theta) \approx \hat{g} = \frac{1}{m} \sum_{i=1}^m \nabla_{\theta} \log P(\tau^{(i)}; \theta) R(\tau^{(i)})$$

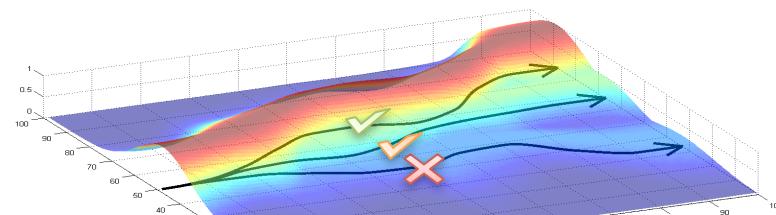
- Valid even when
 - R is discontinuous and/or **unknown**
 - Sample space (of paths) is a discrete set



Likelihood Ratio Gradient: Intuition

$$\nabla U(\theta) \approx \hat{g} = \frac{1}{m} \sum_{i=1}^m \nabla_{\theta} \log P(\tau^{(i)}; \theta) R(\tau^{(i)})$$

- Gradient tries to:
 - Increase probability of paths with positive R
 - Decrease probability of paths with negative R



! Likelihood ratio changes **probabilities** of experienced paths,
does not try to change the paths (<-> Path Derivative)

Let's Decompose Path into States and Actions

$$\nabla_{\theta} \log P(\tau^{(i)}; \theta) = \nabla_{\theta} \log \left[\prod_{t=0}^H \underbrace{P(s_{t+1}^{(i)} | s_t^{(i)}, u_t^{(i)})}_{\text{dynamics model}} \cdot \underbrace{\pi_{\theta}(u_t^{(i)} | s_t^{(i)})}_{\text{policy}} \right]$$

Let's Decompose Path into States and Actions

$$\begin{aligned}\nabla_{\theta} \log P(\tau^{(i)}; \theta) &= \nabla_{\theta} \log \left[\prod_{t=0}^H \underbrace{P(s_{t+1}^{(i)} | s_t^{(i)}, u_t^{(i)})}_{\text{dynamics model}} \cdot \underbrace{\pi_{\theta}(u_t^{(i)} | s_t^{(i)})}_{\text{policy}} \right] \\ &= \nabla_{\theta} \left[\sum_{t=0}^H \log P(s_{t+1}^{(i)} | s_t^{(i)}, u_t^{(i)}) + \sum_{t=0}^H \log \pi_{\theta}(u_t^{(i)} | s_t^{(i)}) \right]\end{aligned}$$

Let's Decompose Path into States and Actions

$$\begin{aligned}\nabla_{\theta} \log P(\tau^{(i)}; \theta) &= \nabla_{\theta} \log \left[\prod_{t=0}^H \underbrace{P(s_{t+1}^{(i)} | s_t^{(i)}, u_t^{(i)})}_{\text{dynamics model}} \cdot \underbrace{\pi_{\theta}(u_t^{(i)} | s_t^{(i)})}_{\text{policy}} \right] \\ &= \nabla_{\theta} \left[\sum_{t=0}^H \log P(s_{t+1}^{(i)} | s_t^{(i)}, u_t^{(i)}) + \sum_{t=0}^H \log \pi_{\theta}(u_t^{(i)} | s_t^{(i)}) \right] \\ &= \nabla_{\theta} \sum_{t=0}^H \log \pi_{\theta}(u_t^{(i)} | s_t^{(i)})\end{aligned}$$

Let's Decompose Path into States and Actions

$$\begin{aligned}\nabla_{\theta} \log P(\tau^{(i)}; \theta) &= \nabla_{\theta} \log \left[\prod_{t=0}^H \underbrace{P(s_{t+1}^{(i)} | s_t^{(i)}, u_t^{(i)})}_{\text{dynamics model}} \cdot \underbrace{\pi_{\theta}(u_t^{(i)} | s_t^{(i)})}_{\text{policy}} \right] \\ &= \nabla_{\theta} \left[\sum_{t=0}^H \log P(s_{t+1}^{(i)} | s_t^{(i)}, u_t^{(i)}) + \sum_{t=0}^H \log \pi_{\theta}(u_t^{(i)} | s_t^{(i)}) \right] \\ &= \nabla_{\theta} \sum_{t=0}^H \log \pi_{\theta}(u_t^{(i)} | s_t^{(i)}) \\ &= \sum_{t=0}^H \underbrace{\nabla_{\theta} \log \pi_{\theta}(u_t^{(i)} | s_t^{(i)})}_{\text{no dynamics model required!!}}\end{aligned}$$

Likelihood Ratio Gradient Estimate

The following expression provides us with an unbiased estimate of the gradient, and we can compute it without access to a dynamics model:

$$\hat{g} = \frac{1}{m} \sum_{i=1}^m \nabla_{\theta} \log P(\tau^{(i)}; \theta) R(\tau^{(i)})$$

Here:

$$\nabla_{\theta} \log P(\tau^{(i)}; \theta) = \sum_{t=0}^H \underbrace{\nabla_{\theta} \log \pi_{\theta}(u_t^{(i)} | s_t^{(i)})}_{\text{no dynamics model required!!}}$$

Unbiased means:

$$\mathbb{E}[\hat{g}] = \nabla_{\theta} U(\theta)$$

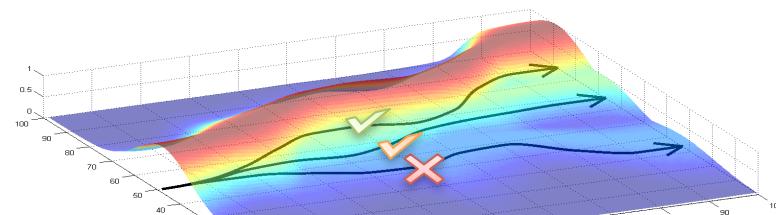
Likelihood Ratio Gradient Estimate

- As formulated thus far: unbiased but very noisy
- Fixes that lead to real-world practicality
 - Baseline
 - Temporal structure
- Also: KL-divergence trust region / natural gradient (= general trick, equally applicable to perturbation analysis and finite differences)

Likelihood Ratio Gradient: Intuition

$$\nabla U(\theta) \approx \hat{g} = \frac{1}{m} \sum_{i=1}^m \nabla_{\theta} \log P(\tau^{(i)}; \theta) R(\tau^{(i)})$$

- Gradient tries to:
 - Increase probability of paths with positive R
 - Decrease probability of paths with negative R



! Likelihood ratio changes probabilities of experienced paths,
does not try to change the paths (<-> Path Derivative)

Likelihood Ratio Gradient: Baseline

$$\nabla U(\theta) \approx \hat{g} = \frac{1}{m} \sum_{i=1}^m \nabla_\theta \log P(\tau^{(i)}; \theta) R(\tau^{(i)})$$

→ Consider baseline b: $\nabla U(\theta) \approx \hat{g} = \frac{1}{m} \sum_{i=1}^m \nabla_\theta \log P(\tau^{(i)}; \theta) (R(\tau^{(i)}) - b)$

$$\mathbb{E} [\nabla_\theta \log P(\tau; \theta) b]$$

$$= \sum_{\tau} P(\tau; \theta) \nabla_\theta \log P(\tau; \theta) b$$

$$= \sum_{\tau} P(\tau; \theta) \frac{\nabla_\theta P(\tau; \theta)}{P(\tau; \theta)} b$$

$$= \sum_{\tau} \nabla_\theta P(\tau; \theta) b$$

$$= \nabla_\theta \left(\sum_{\tau} P(\tau) b \right) = b \nabla_\theta \left(\sum_{\tau} P(\tau) \right) = b \times 0 \quad \blacksquare$$

still unbiased!

[Williams 1992]

OK as long as baseline doesn't depend on action
in logprob(action)

Likelihood Ratio and Temporal Structure

- **Current estimate:**
$$\begin{aligned}\hat{g} &= \frac{1}{m} \sum_{i=1}^m \nabla_{\theta} \log P(\tau^{(i)}; \theta)(R(\tau^{(i)}) - b) \\ &= \frac{1}{m} \sum_{i=1}^m \left(\sum_{t=0}^{H-1} \nabla_{\theta} \log \pi_{\theta}(u_t^{(i)} | s_t^{(i)}) \right) \left(\sum_{t=0}^{H-1} R(s_t^{(i)}, u_t^{(i)}) - b \right) \\ &= \frac{1}{m} \sum_{i=1}^m \left(\sum_{t=0}^{H-1} \nabla_{\theta} \log \pi_{\theta}(u_t^{(i)} | s_t^{(i)}) \left[\left(\sum_{k=0}^{t-1} R(s_k^{(i)}, u_k^{(i)}) \right) + \left(\sum_{k=t}^{H-1} R(s_k^{(i)}, u_k^{(i)}) \right) - b \right] \right)\end{aligned}$$

Doesn't depend on $u_t^{(i)}$

Ok to depend on $s_t^{(i)}$

- Removing terms that don't depend on current action can lower variance:

$$\frac{1}{m} \sum_{i=1}^m \sum_{t=0}^{H-1} \nabla_{\theta} \log \pi_{\theta}(u_t^{(i)} | s_t^{(i)}) \left(\sum_{k=t}^{H-1} R(s_k^{(i)}, u_k^{(i)}) - b(s_t^{(i)}) \right)$$

Baseline Choices

- Good choice for b ?

- Constant baseline: $b = \mathbb{E}[R(\tau)] \approx \frac{1}{m} \sum_{i=1}^m R(\tau^{(i)})$
- Optimal Constant baseline: $b = \frac{\sum_i (\nabla_\theta \log P(\tau^{(i)}; \theta))^2 R(\tau^{(i)})}{\sum_i (\nabla_\theta \log P(\tau^{(i)}; \theta))^2}$
- Time-dependent baseline: $b_t = \frac{1}{m} \sum_{i=1}^m \sum_{k=t}^{H-1} R(s_k^{(i)}, u_k^{(i)})$
- State-dependent expected return:

$$b(s_t) = \mathbb{E}[r_t + r_{t+1} + r_{t+2} + \dots + r_{H-1}] = V^\pi(s_t)$$

→ Increase logprob of action proportionally to how much its returns are better than the expected return under the current policy

Estimation of V^π

$$\frac{1}{m} \sum_{i=1}^m \sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(u_t^{(i)} | s_t^{(i)}) \left(\sum_{k=t}^{H-1} R(s_k^{(i)}, u_k^{(i)}) - \underbrace{V^\pi(s_k^{(i)})}_{\text{Red bracket}} \right)$$

How to estimate?

- Init $V_{\phi_0}^\pi$
 - Collect trajectories τ_1, \dots, τ_m
 - Regress against empirical return:

$$\phi_{i+1} \leftarrow \arg \min_{\phi} \frac{1}{m} \sum_{i=1}^m \sum_{t=0}^{H-1} \left(V_{\phi}^\pi(s_t^{(i)}) - \left(\sum_{k=t}^{H-1} R(s_k^{(i)}, u_k^{(i)}) \right) \right)^2$$

Estimation of V^π

- Bellman Equation for V^π

$$V^\pi(s) = \sum_u \pi(u|s) \sum_{s'} P(s'|s, u) [R(s, u, s') + \gamma V^\pi(s')]$$

- Init $V_{\phi_0}^\pi$
 - Collect data $\{s, u, s', r\}$
 - Fitted V iteration:

$$\phi_{i+1} \leftarrow \min_{\phi} \sum_{(s, u, s', r)} \|r + V_{\phi_i}^\pi(s') - V_\phi(s)\|_2^2 + \lambda \|\phi - \phi_i\|_2^2$$

Vanilla Policy Gradient

Algorithm 1 “Vanilla” policy gradient algorithm

Initialize policy parameter θ , baseline b

for iteration=1, 2, ... **do**

 Collect a set of trajectories by executing the current policy

 At each timestep in each trajectory, compute

 the *return* $R_t = \sum_{t'=t}^{T-1} \gamma^{t'-t} r_{t'}$, and

 the *advantage estimate* $\hat{A}_t = R_t - b(s_t)$.

 Re-fit the baseline, by minimizing $\|b(s_t) - R_t\|^2$,
 summed over all trajectories and timesteps.

 Update the policy, using a policy gradient estimate \hat{g} ,
 which is a sum of terms $\nabla_\theta \log \pi(a_t | s_t, \theta) \hat{A}_t$

end for

Recall Our Likelihood Ratio PG Estimator

$$\frac{1}{m} \sum_{i=1}^m \sum_{t=0}^{H-1} \nabla_{\theta} \log \pi_{\theta}(u_t^{(i)} | s_t^{(i)}) \left(\sum_{k=t}^{H-1} R(s_k^{(i)}, u_k^{(i)}) - V^{\pi}(s_k^{(i)}) \right)$$

Recall Our Likelihood Ratio PG Estimator

$$\frac{1}{m} \sum_{i=1}^m \sum_{t=0}^{H-1} \nabla_{\theta} \log \pi_{\theta}(u_t^{(i)} | s_t^{(i)}) \left(\sum_{k=t}^{H-1} R(s_k^{(i)}, u_k^{(i)}) - V^{\pi}(s_k^{(i)}) \right)$$

Recall Our Likelihood Ratio PG Estimator

$$\frac{1}{m} \sum_{i=1}^m \sum_{t=0}^{H-1} \nabla_{\theta} \log \pi_{\theta}(u_t^{(i)} | s_t^{(i)}) \left(\sum_{k=t}^{H-1} R(s_k^{(i)}, u_k^{(i)}) - V^{\pi}(s_k^{(i)}) \right)$$

- Estimation of Q from *single* roll-out

$$Q^{\pi}(s, u) = \mathbb{E}[r_0 + r_1 + r_2 + \dots | s_0 = s, a_0 = a]$$

Recall Our Likelihood Ratio PG Estimator

$$\frac{1}{m} \sum_{i=1}^m \sum_{t=0}^{H-1} \nabla_{\theta} \log \pi_{\theta}(u_t^{(i)} | s_t^{(i)}) \left(\sum_{k=t}^{H-1} R(s_k^{(i)}, u_k^{(i)}) - V^{\pi}(s_k^{(i)}) \right)$$



- Estimation of Q from *single* roll-out

$$Q^{\pi}(s, u) = \mathbb{E}[r_0 + r_1 + r_2 + \dots | s_0 = s, a_0 = a]$$

- = high variance per sample based / no generalization

Further Refinements

$$\frac{1}{m} \sum_{i=1}^m \sum_{t=0}^{H-1} \nabla_{\theta} \log \pi_{\theta}(u_t^{(i)} | s_t^{(i)}) \left(\sum_{k=t}^{H-1} R(s_k^{(i)}, u_k^{(i)}) - V^{\pi}(s_k^{(i)}) \right)$$

- Estimation of Q from *single* roll-out

$$Q^{\pi}(s, u) = \mathbb{E}[r_0 + r_1 + r_2 + \dots | s_0 = s, a_0 = a]$$

- = high variance per sample based / no generalization
 - Reduce variance by discounting

Recall Our Likelihood Ratio PG Estimator

$$\frac{1}{m} \sum_{i=1}^m \sum_{t=0}^{H-1} \nabla_{\theta} \log \pi_{\theta}(u_t^{(i)} | s_t^{(i)}) \left(\sum_{k=t}^{H-1} R(s_k^{(i)}, u_k^{(i)}) - V^{\pi}(s_k^{(i)}) \right)$$



- Estimation of Q from *single* roll-out

$$Q^{\pi}(s, u) = \mathbb{E}[r_0 + r_1 + r_2 + \dots | s_0 = s, a_0 = a]$$

- = high variance per sample based / no generalization
 - Reduce variance by discounting
 - Reduce variance by function approximation (=critic)

Variance Reduction by Discounting

$$Q^\pi(s, u) = \mathbb{E}[r_0 + r_1 + r_2 + \dots | s_0 = s, a_0 = a]$$

→ introduce discount factor as a hyperparameter to improve estimate of Q:

$$Q^{\pi, \gamma}(s, u) = \mathbb{E}[r_0 + \gamma r_1 + \gamma^2 r_2 + \dots | s_0 = s, a_0 = a]$$

Reducing Variance by Function Approximation

$$Q^{\pi, \gamma}(s, u) = \mathbb{E}[r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots \mid s_0 = s, u_0 = u]$$

Reducing Variance by Function Approximation

$$\begin{aligned} Q^{\pi, \gamma}(s, u) &= \mathbb{E}[r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots \mid s_0 = s, u_0 = u] \\ &= \mathbb{E}[r_0 + \gamma \tilde{V}^{\pi}(s_1) \mid s_0 = s, u_0 = u] \end{aligned}$$

Reducing Variance by Function Approximation

$$\begin{aligned} Q^{\pi, \gamma}(s, u) &= \mathbb{E}[r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots \mid s_0 = s, u_0 = u] \\ &= \mathbb{E}[r_0 + \gamma V^\pi(s_1) \mid s_0 = s, u_0 = u] \\ &= \mathbb{E}[r_0 + \gamma r_1 + \gamma^2 V^\pi(s_2) \mid s_0 = s, u_0 = u] \\ &\quad \vdots \end{aligned}$$

Reducing Variance by Function Approximation

$$\begin{aligned} Q^{\pi, \gamma}(s, u) &= \mathbb{E}[r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots \mid s_0 = s, u_0 = u] \\ &= \mathbb{E}[r_0 + \gamma V^\pi(s_1) \mid s_0 = s, u_0 = u] \\ &= \mathbb{E}[r_0 + \gamma r_1 + \gamma^2 V^\pi(s_2) \mid s_0 = s, u_0 = u] \\ &= \mathbb{E}[r_0 + \gamma r_1 + \gamma^2 r_2 + \gamma^3 V^\pi(s_3) \mid s_0 = s, u_0 = u] \\ &= \cdots \end{aligned}$$

- **Async Advantage Actor Critic (A3C) [Mnih et al, 2016]**
 - \hat{Q} one of the above choices (e.g. k=5 step lookahead)

Reducing Variance by Function Approximation

$$\begin{aligned} Q^{\pi, \gamma}(s, u) &= \mathbb{E}[r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots \mid s_0 = s, u_0 = u] && (1 - \lambda) \\ &= \mathbb{E}[r_0 + \gamma V^\pi(s_1) \mid s_0 = s, u_0 = u] && (1 - \lambda)\lambda \\ &= \mathbb{E}[r_0 + \gamma r_1 + \gamma^2 V^\pi(s_2) \mid s_0 = s, u_0 = u] && (1 - \lambda)\lambda^2 \\ &= \mathbb{E}[r_0 + \gamma r_1 + \gamma^2 r_2 + \gamma^3 V^\pi(s_3) \mid s_0 = s, u_0 = u] \\ &= \dots && (1 - \lambda)\lambda^3 \end{aligned}$$

- **Generalized Advantage Estimation (GAE) [Schulman et al, ICLR 2016]**
 - \hat{Q} = lambda exponentially weighted average of all the above

Reducing Variance by Function Approximation

$$\begin{aligned} Q^{\pi, \gamma}(s, u) &= \mathbb{E}[r_0 + \gamma r_1 + \gamma^2 r_2 + \dots \mid s_0 = s, u_0 = u] && (1 - \lambda) \\ &= \mathbb{E}[r_0 + \gamma V^\pi(s_1) \mid s_0 = s, u_0 = u] && (1 - \lambda)\lambda \\ &= \mathbb{E}[r_0 + \gamma r_1 + \gamma^2 V^\pi(s_2) \mid s_0 = s, u_0 = u] && (1 - \lambda)\lambda^2 \\ &= \mathbb{E}[r_0 + \gamma r_1 + \gamma^2 r_2 + \gamma^3 V^\pi(s_3) \mid s_0 = s, u_0 = u] \\ &= \dots && (1 - \lambda)\lambda^3 \end{aligned}$$

- **Generalized Advantage Estimation (GAE) [Schulman et al, ICLR 2016]**
 - \hat{Q} = lambda exponentially weighted average of all the above
- \sim TD(lambda) / eligibility traces [Sutton and Barto, 1990]

Actor-Critic with A3C or GAE

■ Policy Gradient + Generalized Advantage Estimation:

- Init π_{θ_0} $V_{\phi_0}^{\pi}$
- Collect roll-outs $\{s, u, s', r\}$ and $\hat{Q}_i(s, u)$
- Update: $\phi_{i+1} \leftarrow \min_{\phi} \sum_{(s,u,s',r)} \|\hat{Q}_i(s, u) - V_{\phi}^{\pi}(s)\|_2^2 + \kappa \|\phi - \phi_i\|_2^2$
$$\theta_{i+1} \leftarrow \theta_i + \alpha \frac{1}{m} \sum_{k=1}^m \sum_{t=0}^{H-1} \nabla_{\theta} \log \pi_{\theta_i}(u_t^{(k)} | s_t^{(k)}) \left(\hat{Q}_i(s_t^{(k)}, u_t^{(k)}) - V_{\phi_i}^{\pi}(s_t^{(k)}) \right)$$

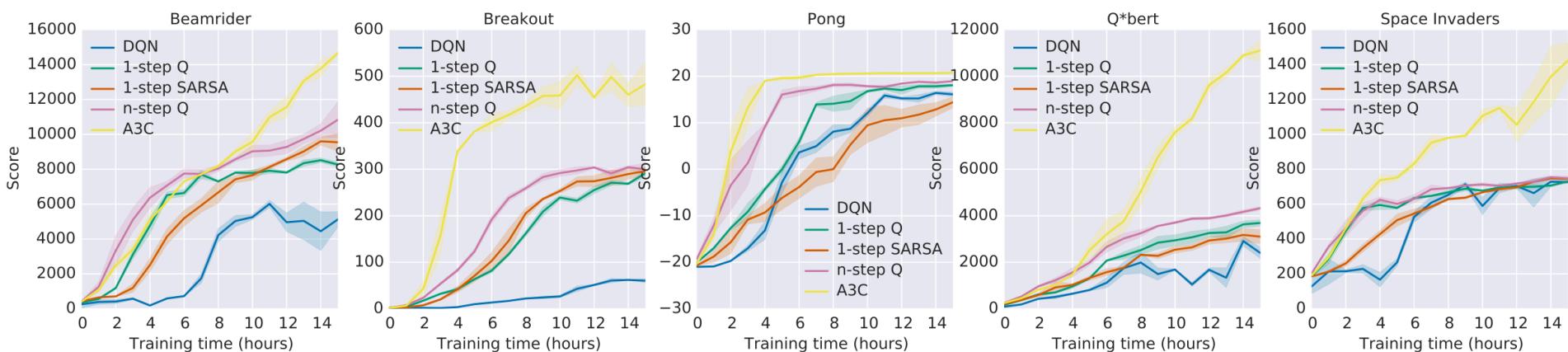
Note: many variations, e.g. could instead use 1-step for V, full roll-out for pi:

$$\phi_{i+1} \leftarrow \min_{\phi} \sum_{(s,u,s',r)} \|r + V_{\phi_i}^{\pi}(s') - V_{\phi}(s)\|_2^2 + \lambda \|\phi - \phi_i\|_2^2$$

$$\theta_{i+1} \leftarrow \theta_i + \alpha \frac{1}{m} \sum_{k=1}^m \sum_{t=0}^{H-1} \nabla_{\theta} \log \pi_{\theta_i}(u_t^{(k)} | s_t^{(k)}) \left(\sum_{t'=t}^{H-1} r_{t'}^{(k)} - V_{\phi_i}^{\pi}(s_{t'}^{(k)}) \right)$$

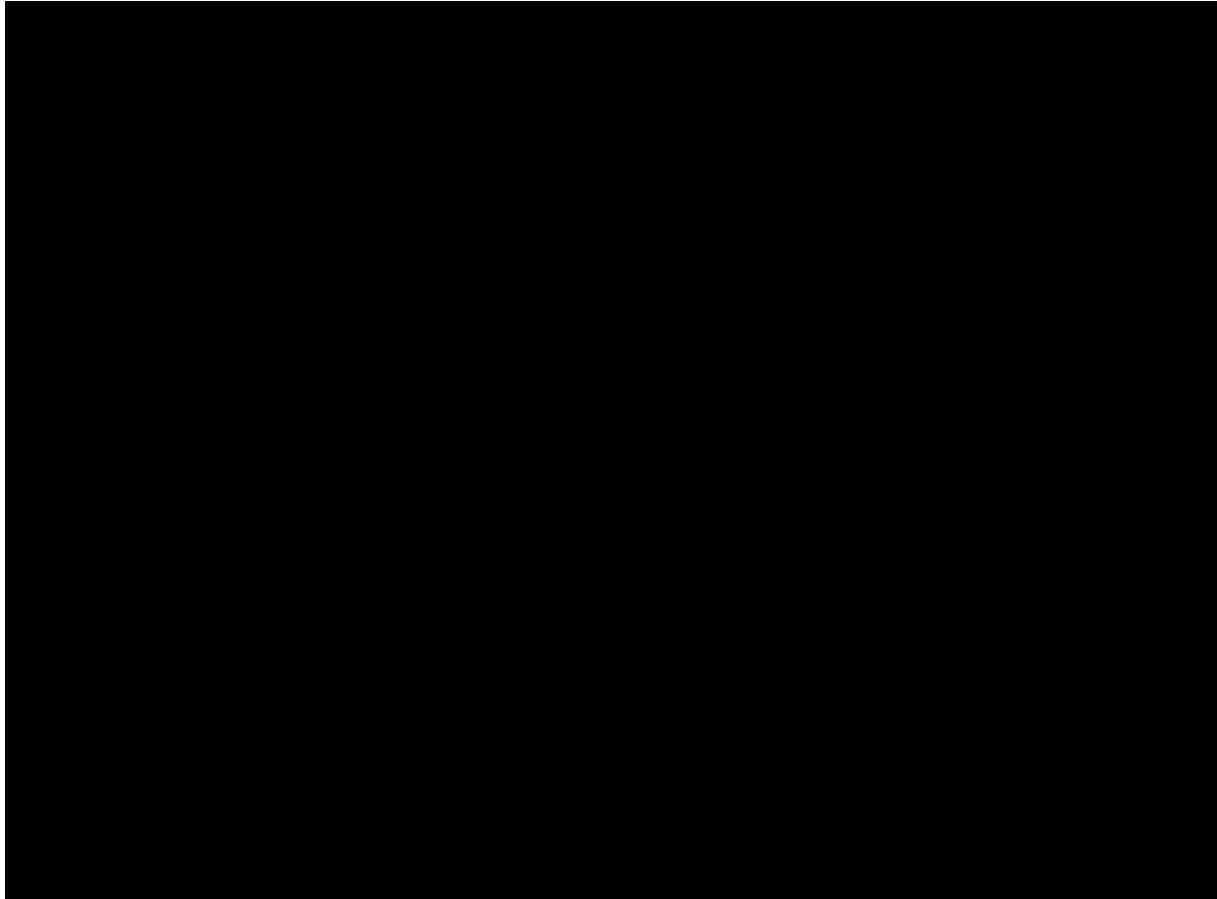
Async Advantage Actor Critic (A3C)

- [Mnih et al, ICML 2016]
 - Likelihood Ratio Policy Gradient
 - n-step Advantage Estimation

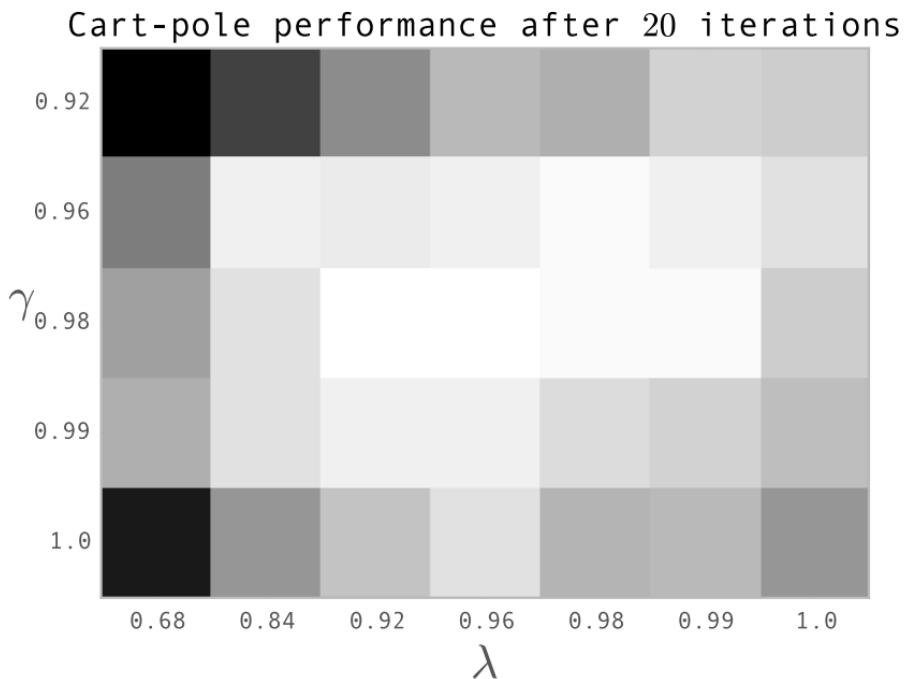
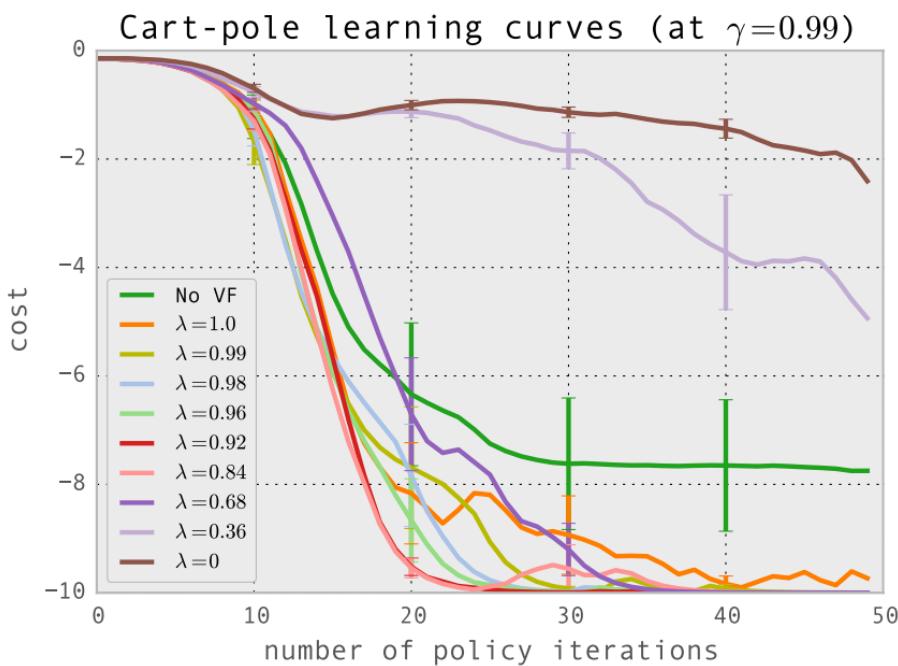




A3C -- labyrinth

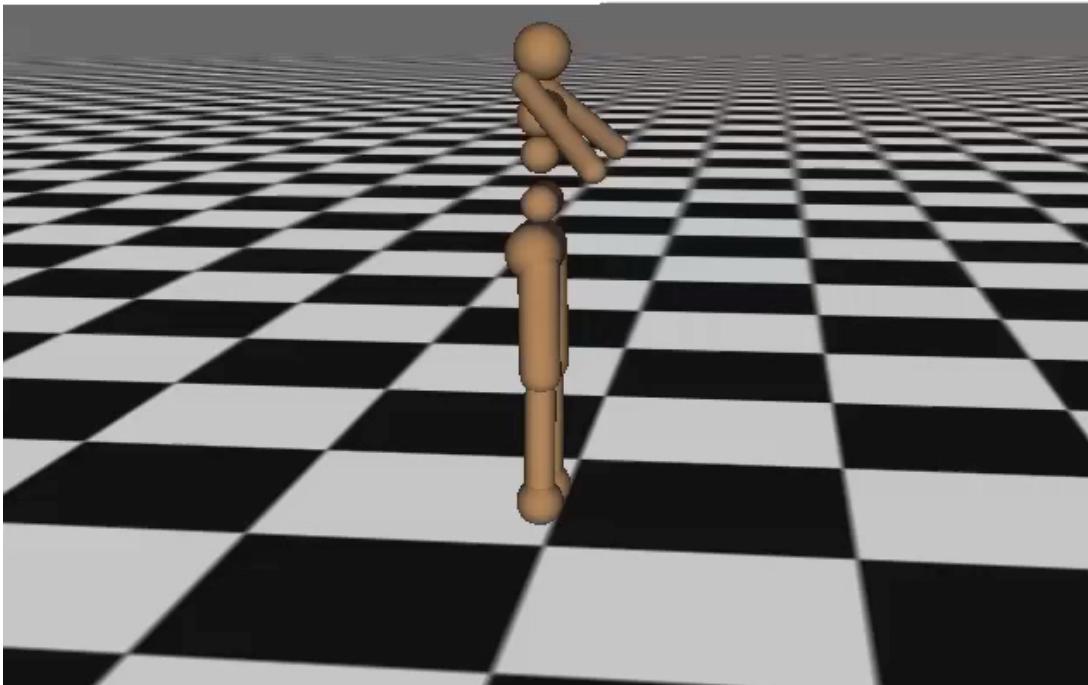


GAE: Effect of gamma and lambda



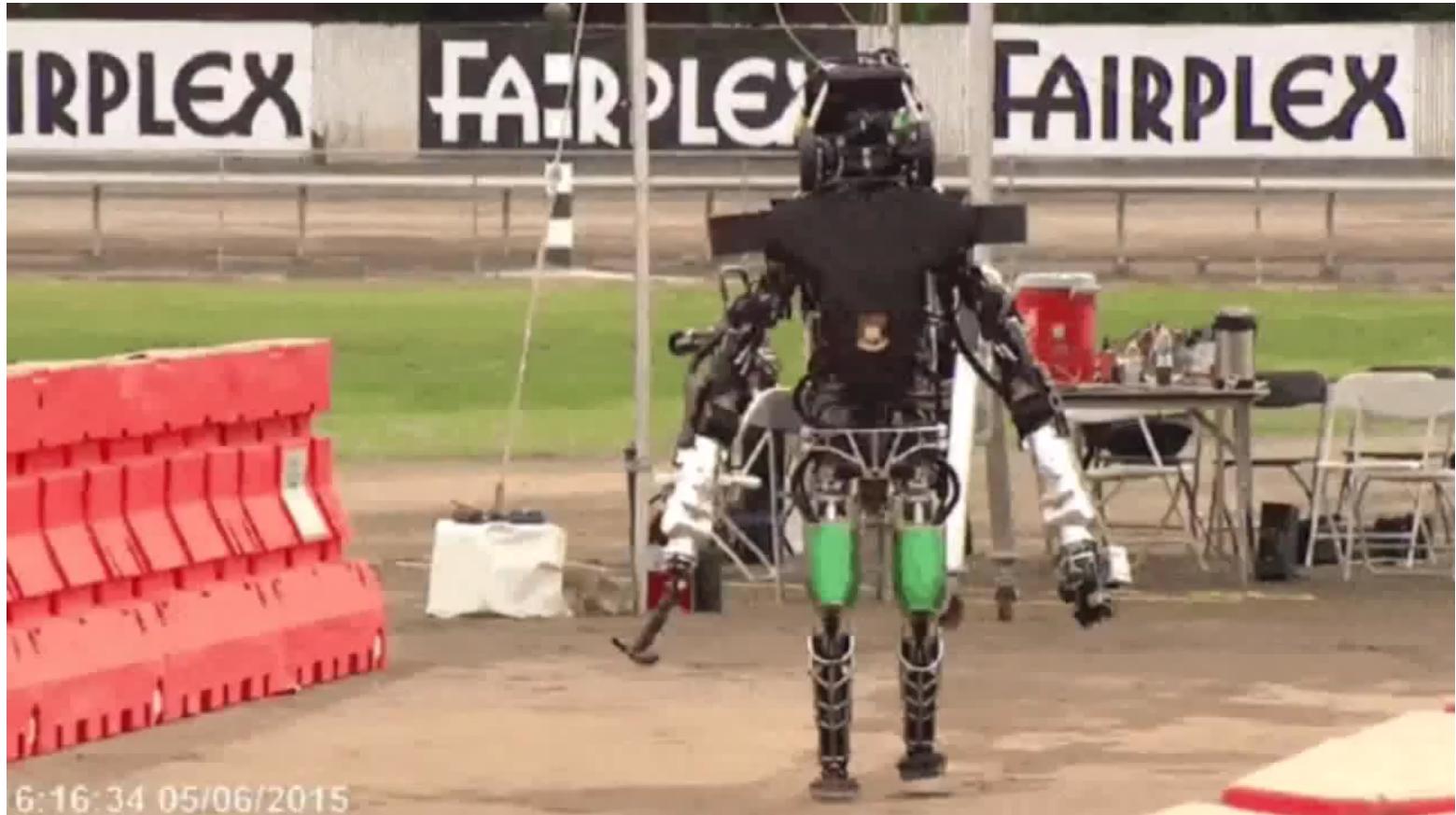
Learning Locomotion (TRPO + GAE)

Iteration 0



[Schulman, Moritz, Levine, Jordan, Abbeel, 2016]

In Contrast: Darpa Robotics Challenge



6:16 34 05/06/2015

