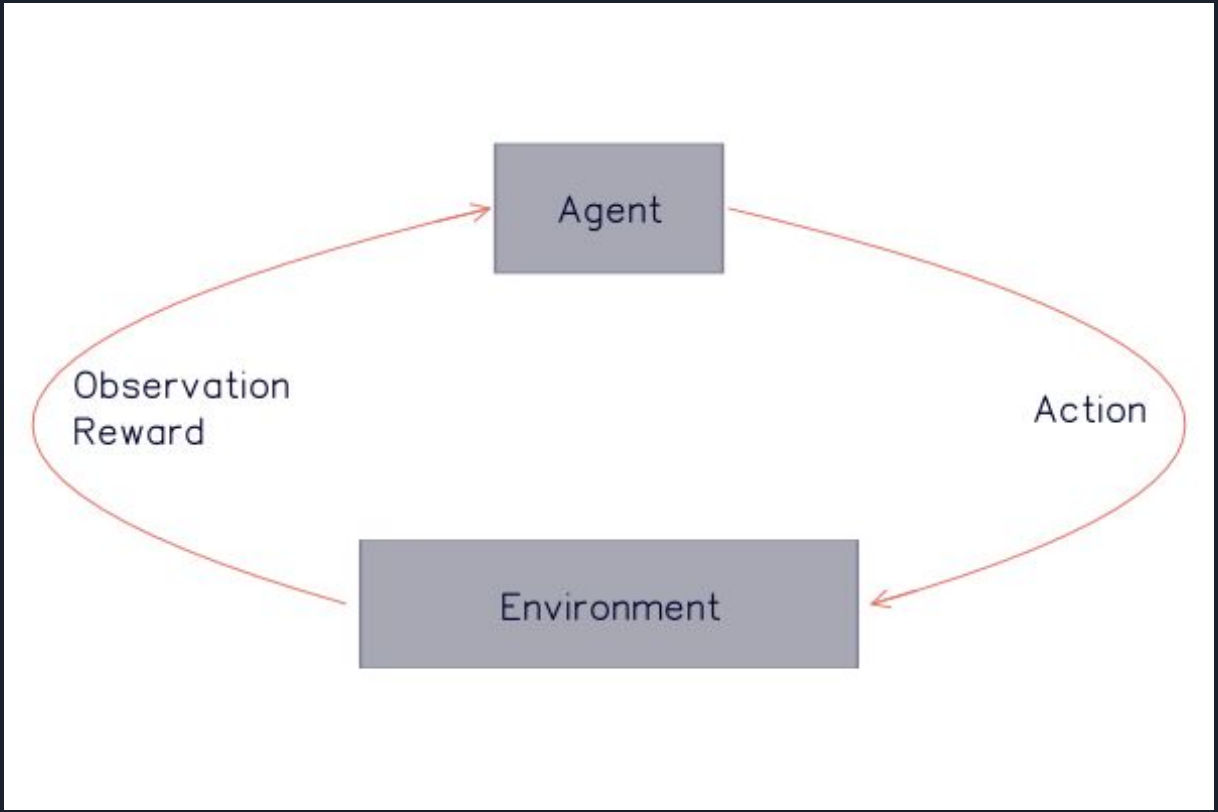
HIGH-DIMENSIONAL CONTINUOUS CONTROL USING GENERALIZED ADVANTAGE ESTIMATION

Paper by John Schulman (and others)

Presentation by Balavivek Sivanantham

Introduction



Advantage function Estimation

- Let V be an approximate value function. But V is not a true value function i.e., the TD residual of V with discount γ. Using V we can derive a class of advantage function estimators as follows

$$
\begin{aligned}
\hat{A}_t^{(1)} &= r_t + \gamma V(s_{t+1}) - V(s_t) \\
\hat{A}_t^{(2)} &= r_t + \gamma r_{t+1} + \gamma^2 V(s_{t+2}) - V(s_t) \\
\cdots &= \cdots \\
\hat{A}_t^{(\infty)} &= r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots - V(s_t)
\end{aligned}
$$

Generalized Advantage Estimator (GAE)

In this paper "**High-Dimensional Continuous Control Using Generalized Advantage Estimation**" uses discounted sum of TD(Temporal Difference) residuals.

$$\delta_t^V = r_t + \gamma V(s_{t+1}) - V(s_t)$$

and compute an estimator of the k-step discounted advantage:

$$\hat{A}_t^{(k)} = \sum_{l=0}^{k-1} \gamma^l \delta_{t+l}^V$$

They define their generalized advantage estimator (GAE) as the weighted average of the advantage estimators above, which reduce to a sum of discounted TD residuals:

$$\hat{A}_t^{GAE(\gamma,\lambda)} = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^V$$

```
advantage = discount(rewards + GAMMA * vpred[1:] * (1 - terminals_array[1:]) - vpred[:-1], GAMMA * LAMBDA)
tdlamret = advantage + np.array(buffer_v)
advantage = (advantage - advantage.mean()) / np.maximum(advantage.std(), 1e-6)
```

Reward Shaping Interpretation

$\Phi : S \rightarrow R$ an arbitrary real-valued function on the state space:

$$\tilde{r}(s, a, s') = r(s, a, s') + \gamma\Phi(s') - \Phi(s)$$

If we try to maximize the sum of ($\gamma\lambda$)-discounted sum of (transformed) rewards and set $\Phi=V$, we get precisely the GAE!

Policy Optimization Algorithm

Initialize policy parameter θ_0 and value function parameter ϕ_0.
for $i = 0, 1, 2, \ldots$ **do**
    Simulate current policy π_{θ_i} until N timesteps are obtained.
    Compute δ_t^V at all timesteps $t \in \{1, 2, \ldots, N\}$, using $V = V_{\phi_i}$.
    Compute $\hat{A}_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^V$ at all timesteps.
    Compute θ_{i+1} with TRPO update, Equation (31).
    Compute ϕ_{i+1} with Equation (30).
end for

Summary

They present and analyze a specific kind of estimator, the GAE, which has a bias-variance "knob" with the λ (and γ, technically). By adjusting the knob, it might be possible to get low variance, low biased estimates, which would drastically improve the sample efficiency of policy gradient methods. They also present a way to estimate the value method using a trust region method. With these components, they are able to achieve high performance on challenging reinforcement learning tasks with continuous control.

Experiment Setup

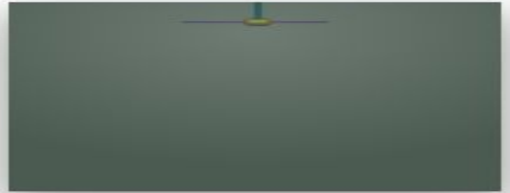
Ant-v2
Make a 3D four-legged robot walk.

HalfCheetah-v2
Make a 2D cheetah robot run.

Hopper-v2
Make a 2D robot hop.

Humanoid-v2
Make a 3D two-legged robot walk.

HumanoidStandup-v2
Make a 3D two-legged robot standup.



InvertedDoublePendulum-v2
Balance a pole on a pole on a cart.

Task Details

- For **Cart-Pole Balancing** task, 20 trajectories per batch, with maximum length of 1000 timesteps.
- The Simulated robot task were simulated using the MuJoCo physics engine.
    - Humanoid — 33 State Dimension — 10 Actuated Degree of Freedom — 5000 time steps
    - Quadruped — 29 State Dimension — 8 Actuated Degree of Freedom — 200000 time steps
- Each episode was terminated after 2000 timesteps, if the robot has not reached a terminal state beforehand.

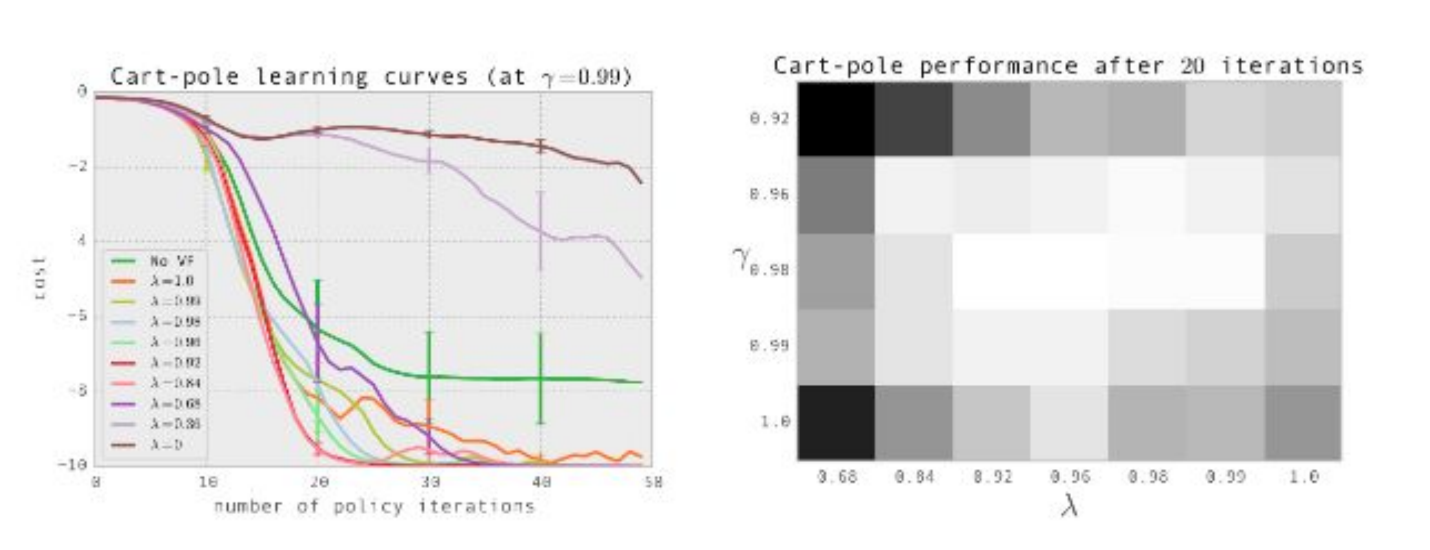
The reward functions are provided in the table below.

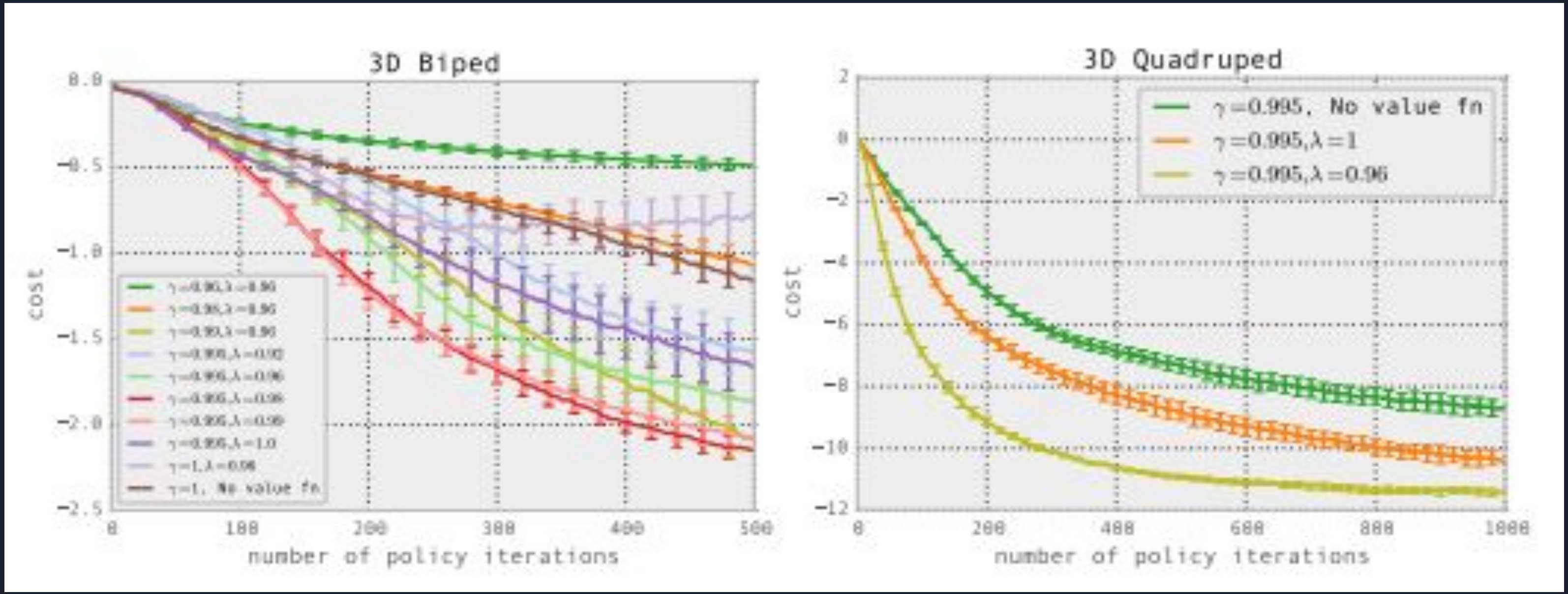
Task	Reward
3D biped locomotion	$v_{\mathrm{fwd}} - 10^{-5}\|u\|^2 - 10^{-5}\|f_{\mathrm{impact}}\|^2 + 0.2$
Quadruped locomotion	$v_{\mathrm{fwd}} - 10^{-6}\|u\|^2 - 10^{-3}\|f_{\mathrm{impact}}\|^2 + 0.05$
Biped getting up	$-(h_{\mathrm{head}} - 1.5)^2 - 10^{-5}\|u\|^2$

Here, $v_{\mathrm{fwd}} :=$ forward velocity, $u :=$ vector of joint torques, $f_{\mathrm{impact}} :=$ impact forces, $h_{\mathrm{head}} :=$ height of the head.

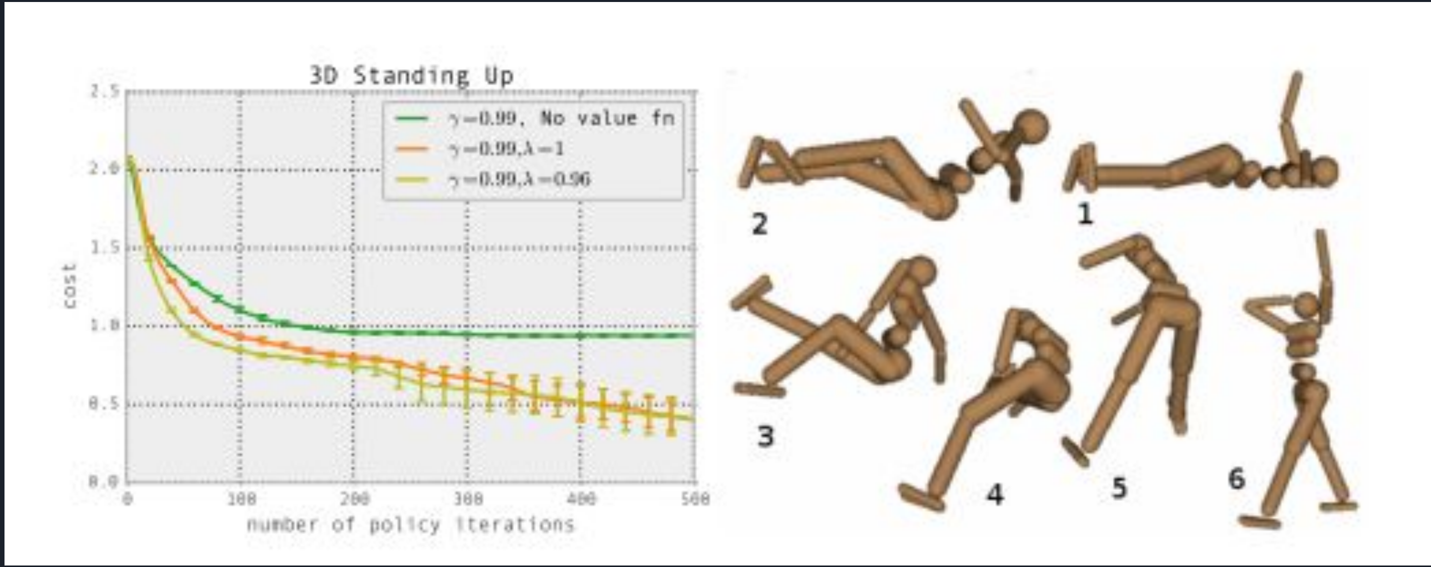
Experiment Results

- Cart-Pole



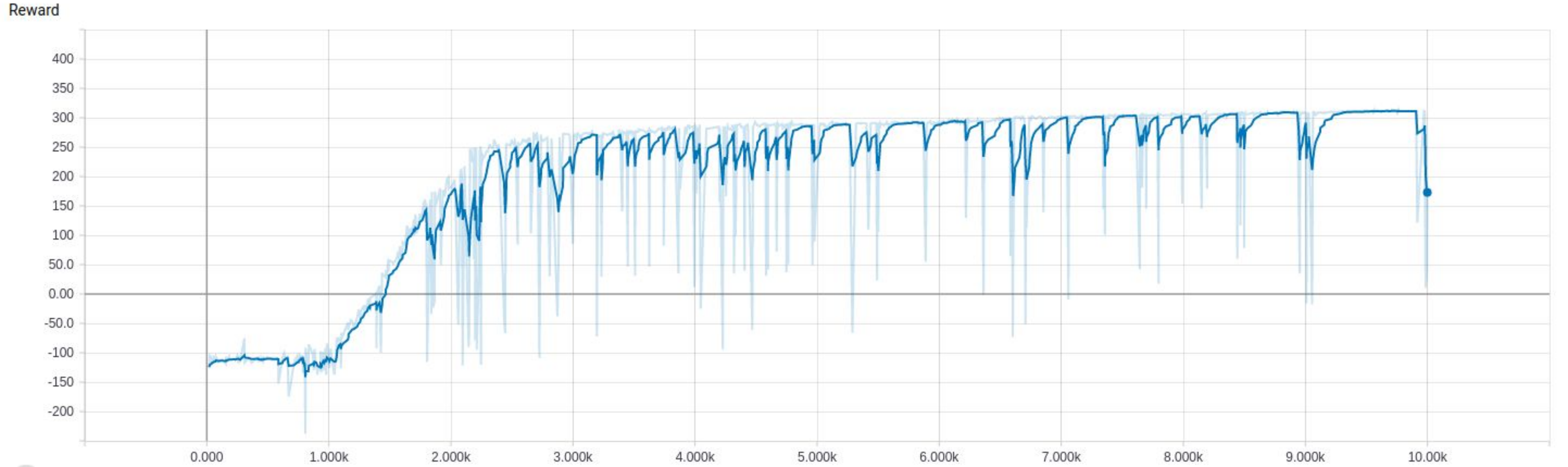

3D Biped
cost
number of policy iterations
3D Quadruped
$\gamma = 0.995$, No value fn
$\gamma = 0.995, \lambda = 1$
$\gamma = 0.995, \lambda = 0.96$
cost
number of policy iterations

- OTHER 3D ROBOT TASKS




[C]ontact forces On
Referenc[e] frames On
T[r]ansparent Off
Display [M]ocap bodies On
Stop [Space]
Advance simulation by one step [right arrow]
[H]ide Menu
Record [V]ideo (Off)
Cap[t]ure frame
Start [i]pdb
Toggle geomgroup visibility 0-4
FPS 31
Solver iterations 4

BipedalWalker-v2 solved using PPO with a LSTM layer



Thank you

Questions ?