HIGH-DIMENSIONAL CONTINUOUS CONTROL USING GENERALIZED ADVANTAGE ESTIMATION

John Schulman, 2016

Junyeob Baek

0. ABSTRACT

The two main challanges..(of the policy gradient method for RL)

1. the large number of samples typically required.('cause of high-sample-complexity)

2. the difficulty of obtainning stable and steady improvement despite the nonstationarity of the incoming data.

0. ABSTRACT

The two main challanges..(of the policy gradient method for RL)

1. the large number of samples typically required.('cause of high-sample-complexity)
   -> reduce the variance of policy gradient estimates, with an exponetially-weighted estimator of advantage funtion

2. the difficulty of obtainning stable and steady improvement despite the nonstationarity of the incoming data.
   -> trust region optimization procedure for both the policy and the value funtion

0. ABSTRACT

Results..

1. Strong empirical results on highly challenging 3D locomotion tasks.

   (learning gaits for bipedal and quadrupedal and standing up motion for biped)

   https://sites.google.com/site/gaepapersupp/

2. Fully model-free

   Prior works: hand-crafted policy representations

   Ours: directly map from raw kinematics to joint torques.

1. INTRODUCTION

History..

- (Williams, 1992; Sutton et al., 1999; Baxter & Bartlett, 2000)

  with a parameterized stochastic policy

  It's possible to obtain an Unbiased estimate of the gradient of the expected total returns

1. INTRODUCTION

History..

- (Williams, 1992; Sutton et al., 1999; Baxter & Bartlett, 2000)

  with a parameterized stochastic policy

  It's possible to obtain an Unbiased estimate of the gradient of the expected total returns

  -> Unfortunatley, the variance scales with the time horizon, since the effect of an action is confounded with the effects of past and future actions.

1. INTRODUCTION

History..

- (Williams, 1992; Sutton et al., 1999; Baxter & Bartlett, 2000)

  with a parameterized stochastic policy

  It's possible to obtain an Unbiased estimate of the gradient of the expected total returns

  -> Unfortunatley, the variance scales with the time horizon, since the effect of an action is confounded with the effects of past and future actions.

- (Konda & Tsitsiklis, 2003; Hafner & Riedmiller, 2011)

  Actor-critic method use a value function rather than the empirical returns, obtaining an estimator with lower variance at the cost of introducing bias.

1. INTRODUCTION

History..

- (Williams, 1992; Sutton et al., 1999; Baxter & Bartlett, 2000)

  with a parameterized stochastic policy

  It's possible to obtain an Unbiased estimate of the gradient of the expected total returns

  -> Unfortunatley, the variance scales with the time horizon, since the effect of an action is confounded with the effects of past and future actions.

- (Konda & Tsitsiklis, 2003; Hafner & Riedmiller, 2011)

  Actor-critic method use a value function rather than the empirical returns, obtaining an estimator with lower variance at the cost of introducing bias

  -> But high variance necessitates using more samples and bias is more pernicious.

  bias can cause the algorithm to fail to converge, or to converge to a poor solution

1. INTRODUCTION

We propose..

a family of policy gradient estimators that significantly reduce variance while maintaining a tolerable level of bias

, that is called **the generalized advantage estimator(GAE).**

The contributions of this paper are summarized as follows:

1. An effective variance reduction scheme for policy gradients, GAE.
2. The use of trust region optimization method for the value funtion.
3. By combining above, we obtain an effective learning algorithm for neural network policies.

   -> the results extend the state of the art in using RL for high-dimensional continuous control.

2. PRELIMINARIES

We consider...

an undiscounted formulation of the policy optimization problem.

initial state s_0 is sampled from distribution ρ_0.
sampling actions according to policy $a_t \sim \pi(a_t|s_t)$.
sampling states according to the dynamics $s_{t+1} \sim P(s_{t+1}|s_t, a_t)$.
and a trajectory $(s_0, a_0, s_1, a_1, \ldots)$ is generated by above until terminal state is reached.
a reward $r_t = r(s_t, a_t, s_{t+1})$is reached at each timestep.

The goal is to maximize the expected total reward $\sum_{t=0}^{\infty} r_t$

And the policy gradient methods maximize above by repeatedly estimating the gradient $g := \nabla\theta\mathbb{E}[\sum_{t=0}^{\infty} r_t]$

2. PRELIMINARIES

Several different related expressions for the policy gradient,

$$g = \mathbb{E}\left[\sum_{t=0}^{\infty} \Psi_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\right], \tag{1}$$

where Ψ_t may be one of the following:

1. $\sum_{t=0}^{\infty} r_t$: total reward of the trajectory.
2. $\sum_{t'=t}^{\infty} r_{t'}$: reward following action a_t.
3. $\sum_{t'=t}^{\infty} r_{t'} - b(s_t)$: baselined version of previous formula.
4. $Q^\pi(s_t, a_t)$: state-action value function.
5. $A^\pi(s_t, a_t)$: advantage function.
6. $r_t + V^\pi(s_{t+1}) - V^\pi(s_t)$: TD residual.

The latter formulas use the definitions

$$V^\pi(s_t) := \mathbb{E}_{\substack{s_{t+1:\infty},\\ a_{t:\infty}}}\left[\sum_{l=0}^{\infty} r_{t+l}\right] \qquad Q^\pi(s_t, a_t) := \mathbb{E}_{\substack{s_{t+1:\infty},\\ a_{t+1:\infty}}}\left[\sum_{l=0}^{\infty} r_{t+l}\right] \tag{2}$$

$$A^\pi(s_t, a_t) := Q^\pi(s_t, a_t) - V^\pi(s_t), \quad \text{(Advantage function)}. \tag{3}$$

2. PRELIMINARIES

The choice $\psi_t = A^\pi(s_t, a_t)$

yields almost the lowest possible variance, though in practice,

the advantage function is not known and must be estimated.

By it's definition $A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$

, measures whether or not the action is better or worse than the policy's default behavior.

Hence, we should *choose Ψt to be the advantage function Aπ (st, at),*

for a more rigorous analysis of

the variance of policy gradient estimators and the effect of using a baseline.

2. PRELIMINARIES

A parameter γ

, that allows us to reduce variance by downweighting rewards corresponding to delayed effects at the cost of introducing bias.

This parameter corresponds to the discount factor used in discounted formulations of MDPs, but we treat it **as a variance reduction parameter** in an undiscounted problem.

$$V^{\pi,\gamma}(s_t) := \mathbb{E}_{\substack{s_{t+1:\infty},\\ a_{t:\infty}}}\left[\sum_{l=0}^{\infty}\gamma^l r_{t+l}\right] \qquad Q^{\pi,\gamma}(s_t,a_t) := \mathbb{E}_{\substack{s_{t+1:\infty},\\ a_{t+1:\infty}}}\left[\sum_{l=0}^{\infty}\gamma^l r_{t+l}\right] \tag{4}$$

$$A^{\pi,\gamma}(s_t,a_t) := Q^{\pi,\gamma}(s_t,a_t) - V^{\pi,\gamma}(s_t). \tag{5}$$

The discounted approximation to the policy gradient is defined as follows:

$$g^{\gamma} := \mathbb{E}_{\substack{s_{0:\infty}\\ a_{0:\infty}}}\left[\sum_{t=0}^{\infty} A^{\pi,\gamma}(s_t,a_t)\nabla_\theta \log \pi_\theta(a_t \mid s_t)\right]. \tag{6}$$

2. PRELIMINARIES

γ - just estimator of the advantage funtion,

which is an estimator that does not introduce bias when we use it in place of $A^{\pi,\gamma}$ in Equation (6).

$$g^\gamma := \mathbb{E}_{\substack{s_{0:\infty} \\ a_{0:\infty}}} \left[\sum_{t=0}^{\infty} A^{\pi,\gamma}(s_t, a_t) \nabla_\theta \log \pi_\theta(a_t \mid s_t) \right]. \tag{6}$$

And consider an advantage estimator $\hat{A}_t(s_{0:\infty}, a_{0:\infty})$

, which may in general be a function of the entire trajectory

Definition 1. *The estimator $\hat{A}_t$ is γ-just if*

$$\mathbb{E}_{\substack{s_{0:\infty} \\ a_{0:\infty}}} \left[\hat{A}_t(s_{0:\infty}, a_{0:\infty}) \nabla_\theta \log \pi_\theta(a_t \mid s_t) \right] = \mathbb{E}_{\substack{s_{0:\infty} \\ a_{0:\infty}}} \left[A^{\pi,\gamma}(s_t, a_t) \nabla_\theta \log \pi_\theta(a_t \mid s_t) \right]. \tag{7}$$

It follows immediately that if $\hat{A}_t$ is γ-just for all t, then

$$\mathbb{E}_{\substack{s_{0:\infty} \\ a_{0:\infty}}} \left[\sum_{t=0}^{\infty} \hat{A}_t(s_{0:\infty}, a_{0:\infty}) \nabla_\theta \log \pi_\theta(a_t \mid s_t) \right] = g^\gamma \tag{8}$$

2. PRELIMINARIES

One sufficient condition for $\hat{A}_t$ to be γ-just

is that Aˆ t decomposes as the difference between two functions Qt and bt,

where Qt can depend on any trajectory variables but gives an unbiased estimator of the γ-discounted Q-function,

and bt is an arbitrary function of the states and actions sampled before at.

Proposition 1. *Suppose that* $\hat{A}_t$ *can be written in the form* $\hat{A}_t(s_{0:\infty}, a_{0:\infty}) = Q_t(s_{t:\infty}, a_{t:\infty}) - b_t(s_{0:t}, a_{0:t-1})$ *such that for all* (s_t, a_t), $\mathbb{E}_{s_{t+1:\infty}, a_{t+1:\infty} \mid s_t, a_t}[Q_t(s_{t:\infty}, a_{t:\infty})] = Q^{\pi,\gamma}(s_t, a_t)$. *Then* $\hat{A}$ *is* γ*-just.*

The proof is provided in Appendix B. It is easy to verify that the following expressions are γ-just advantage estimators for $\hat{A}_t$:

- $\sum_{l=0}^{\infty} \gamma^l r_{t+l}$
- $Q^{\pi,\gamma}(s_t, a_t)$
- $A^{\pi,\gamma}(s_t, a_t)$
- $r_t + \gamma V^{\pi,\gamma}(s_{t+1}) - V^{\pi,\gamma}(s_t)$.

3. ADVANTAGE FUNTION ESTIMATION

we can define the **TD residual** of V with discount factor: $\delta_t^V = r_t + V(s_{t+1}) - V(s_t)$

Then the TD residual can be considered as an estimate of the advantage of the action.

$$\begin{aligned}\mathbb{E}_{s_{t+1}}\left[\delta_t^{V^{\pi,\gamma}}\right] &= \mathbb{E}_{s_{t+1}}\left[r_t + \gamma V^{\pi,\gamma}(s_{t+1}) - V^{\pi,\gamma}(s_t)\right] \\ &= \mathbb{E}_{s_{t+1}}\left[Q^{\pi,\gamma}(s_t, a_t) - V^{\pi,\gamma}(s_t)\right] = A^{\pi,\gamma}(s_t, a_t). \qquad (10)\end{aligned}$$

In fact, if we have the correct value function V = V π,γ,

then it is a γ-just advantage estimator, and in fact, an unbiased estimator of Aπ,γ.

However, this estimator is only γ-just for V = V π,γ, otherwise it will yield biased policy gradient estimates.

3. ADVANTAGE FUNTION ESTIMATION

Consider taking the sum of k of these δ terms, which we will denote by **Aˆ (k)**

$$\hat{A}_t^{(1)} := \delta_t^V = -V(s_t) + r_t + \gamma V(s_{t+1}) \tag{11}$$

$$\hat{A}_t^{(2)} := \delta_t^V + \gamma\delta_{t+1}^V = -V(s_t) + r_t + \gamma r_{t+1} + \gamma^2 V(s_{t+2}) \tag{12}$$

$$\hat{A}_t^{(3)} := \delta_t^V + \gamma\delta_{t+1}^V + \gamma^2\delta_{t+2}^V = -V(s_t) + r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 V(s_{t+3}) \tag{13}$$

$$\hat{A}_t^{(k)} := \sum_{l=0}^{k-1} \gamma^l \delta_{t+l}^V = -V(s_t) + r_t + \gamma r_{t+1} + \cdots + \gamma^{k-1} r_{t+k-1} + \gamma^k V(s_{t+k}) \tag{14}$$

Aˆ (k) t involves a k-step estimate of the returns, minus a baseline term −V(s).

3. ADVANTAGE FUNTION ESTIMATION

$$\hat{A}_t^{(k)} := \sum_{l=0}^{k-1} \gamma^l \delta_{t+l}^V = -V(s_t) + r_t + \gamma r_{t+1} + \cdots + \gamma^{k-1} r_{t+k-1} + \gamma^k V(s_{t+k}) \qquad (14)$$

And we can consider Aˆ (k) to be an estimator of the advantage function,

which is only γ-just when V = V π,γ. However, the bias generally becomes smaller as k → ∞.

Taking k → ∞, we get

$$\hat{A}_t^{(\infty)} = \sum_{l=0}^{\infty} \gamma^l \delta_{t+l}^V = -V(s_t) + \sum_{l=0}^{\infty} \gamma^l r_{t+l}, \qquad (15)$$

which is simply the empirical returns minus the value funtion baseline.

3. ADVANTAGE FUNTION ESTIMATION

The generalized advantage estimator **GAE(γ, λ)**

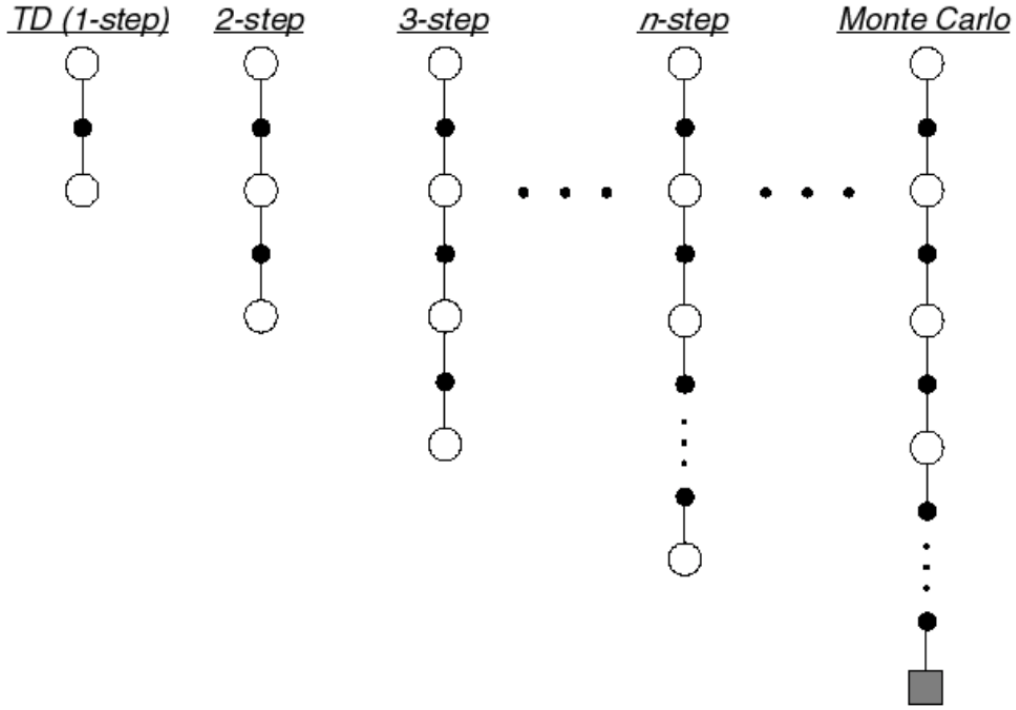
is defined as the exponentially-weighted average of these k-step estimators:

$$
\begin{aligned}
\hat{A}_t^{\mathrm{GAE}(\gamma,\lambda)} &:= (1-\lambda)\left(\hat{A}_t^{(1)} + \lambda \hat{A}_t^{(2)} + \lambda^2 \hat{A}_t^{(3)} + \dots\right) \\
&= (1-\lambda)\left(\delta_t^V + \lambda(\delta_t^V + \gamma\delta_{t+1}^V) + \lambda^2(\delta_t^V + \gamma\delta_{t+1}^V + \gamma^2\delta_{t+2}^V) + \dots\right) \\
&= (1-\lambda)(\delta_t^V(1 + \lambda + \lambda^2 + \dots) + \gamma\delta_{t+1}^V(\lambda + \lambda^2 + \lambda^3 + \dots) \\
&\qquad + \gamma^2\delta_{t+2}^V(\lambda^2 + \lambda^3 + \lambda^4 + \dots) + \dots) \\
&= (1-\lambda)\left(\delta_t^V\left(\frac{1}{1-\lambda}\right) + \gamma\delta_{t+1}^V\left(\frac{\lambda}{1-\lambda}\right) + \gamma^2\delta_{t+2}^V\left(\frac{\lambda^2}{1-\lambda}\right) + \dots\right) \\
&= \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^V
\end{aligned}
\tag{16}
$$

3. ADVANTAGE FUNTION ESTIMATION

TD(λ)..?

- Let TD target look *n* steps into the future



Using TD(n-step),
model can avoid overfitting and biased.

TD(1-step) is the general td-error updates.

...

TD(n-step) is similar to offline Monte Carlo updates.

- Consider the following *n*-step returns for $n = 1, 2, \infty$:

$$
\begin{aligned}
n = 1 \quad (TD) \quad & G_t^{(1)} = R_{t+1} + \gamma V(S_{t+1}) \\
n = 2 \quad & G_t^{(2)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 V(S_{t+2}) \\
\vdots \quad & \vdots \\
n = \infty \quad (MC) \quad & G_t^{(\infty)} = R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{T-1} R_T
\end{aligned}
$$

3. ADVANTAGE FUNTION ESTIMATION

TD(λ)..?

N-step TD updates has benefits of both TD and MC. (parameter: n, step_size)

Hard to find the appropriate hyperparameters..

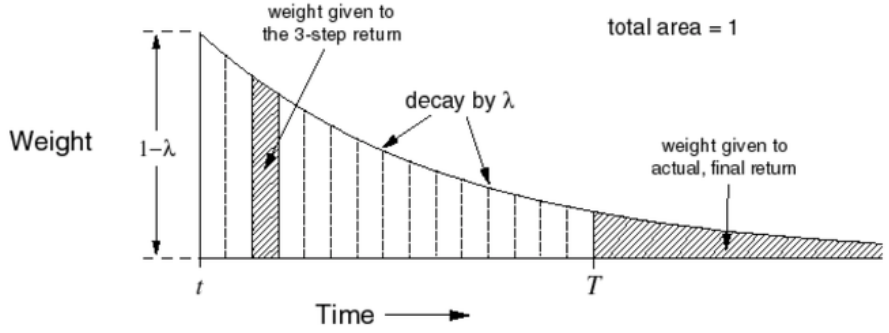
TD(λ) is a geometrically weighted average using weight λ.

TD(λ) has n-step TD's benefits

and also easier than n-step TD to set the hyperparameters.

**

TD(0) : 1-step TD
TD(1) : MC



$$G_t^{\lambda} = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}G_t^{(n)}$$

3. ADVANTAGE FUNTION ESTIMATION

The generalized advantage estimator **GAE(γ, λ)**

is defined as the exponentially-weighted average of these k-step estimators:

$$
\begin{aligned}
\hat{A}_t^{\mathrm{GAE}(\gamma,\lambda)} &:= (1-\lambda)\left(\hat{A}_t^{(1)} + \lambda \hat{A}_t^{(2)} + \lambda^2 \hat{A}_t^{(3)} + \dots\right) \\
&= (1-\lambda)\left(\delta_t^V + \lambda(\delta_t^V + \gamma\delta_{t+1}^V) + \lambda^2(\delta_t^V + \gamma\delta_{t+1}^V + \gamma^2\delta_{t+2}^V) + \dots\right) \\
&= (1-\lambda)(\delta_t^V(1 + \lambda + \lambda^2 + \dots) + \gamma\delta_{t+1}^V(\lambda + \lambda^2 + \lambda^3 + \dots) \\
&\qquad + \gamma^2\delta_{t+2}^V(\lambda^2 + \lambda^3 + \lambda^4 + \dots) + \dots) \\
&= (1-\lambda)\left(\delta_t^V\left(\frac{1}{1-\lambda}\right) + \gamma\delta_{t+1}^V\left(\frac{\lambda}{1-\lambda}\right) + \gamma^2\delta_{t+2}^V\left(\frac{\lambda^2}{1-\lambda}\right) + \dots\right) \\
&= \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^V
\end{aligned}
\tag{16}
$$

3. ADVANTAGE FUNTION ESTIMATION

GAE(γ, 1) is γ-just regardless of the accuracy of V, but it has high variance due to the sum of terms.

GAE(γ, 0) is γ-just for $V = V^{\pi,\gamma}$ and otherwise induces bias, but it typically has much lower variance

.

$$\text{GAE}(\gamma, 0): \quad \hat{A}_t := \delta_t = r_t + \gamma V(s_{t+1}) - V(s_t) \tag{17}$$

$$\text{GAE}(\gamma, 1): \quad \hat{A}_t := \sum_{l=0}^{\infty} \gamma^l \delta_{t+l} = \sum_{l=0}^{\infty} \gamma^l r_{t+l} - V(s_t) \tag{18}$$

The generalized advantage estimator for $0 < \lambda < 1$

makes a **compromise between bias and variance**, controlled by parameter λ.

3. ADVANTAGE FUNTION ESTIMATION

Features..

- An advantage estimator has two separate parameters **γ** and **λ**,
  both of which contribute to the bias-variance tradeoff when using an approximate value function
  .

- **γ** determines the scale of the value function $V^{\pi,\gamma}$, which does not depend on λ.
  Taking γ < 1 introduces bias into the gradient estimate, regardless of the accuracy of V.
  λ < 1 introduces bias only when the value function is inaccurate.

- the best value of λ is much lower than the best value of γ.
  because λ introduces far less bias than γ for a reasonably accurate value function.

4. INTERPRETATION AS REWARD SHAPING

In aspect of ***Reward Shaping…***

Reward shaping (Ng et al., 1999) refers to the following transformation of the reward function of an MDP.

$$\tilde{r}(s, a, s') = r(s, a, s') + \gamma\Phi(s') - \Phi(s), \tag{20}$$

This transformation leaves the policy gradient and optimal policy unchanged,

when our objective is to maximize the discounted sum of rewards.

(In contrast, this paper is concerned with maximizing the undiscounted sum of rewards,

where the discount γ is used as a variance-reduction parameter.)

$$\sum_{l=0}^{\infty} \gamma^l \tilde{r}(s_{t+l}, a_t, s_{t+l+1}) = \sum_{l=0}^{\infty} \gamma^l r(s_{t+l}, a_{t+l}, s_{t+l+1}) - \Phi(s_t). \tag{21}$$

It can be translated by the discounted sum of the original rewards minus a baseline term.

4. INTERPRETATION AS REWARD SHAPING

In aspect of ***Reward Shaping...***

Let's consider using a "steeper" discount γλ, where 0 ≤ λ ≤ 1.

It's easy to see that the shaped reward r˜ equals the Bellman residual term δ V

$$\sum_{l=0}^{\infty}(\gamma\lambda)^l\tilde{r}(s_{t+l},a_t,s_{t+l+1}) = \sum_{l=0}^{\infty}(\gamma\lambda)^l\delta_{t+l}^{V} = \hat{A}_t^{\mathrm{GAE}(\gamma,\lambda)}. \tag{25}$$

By considering the γλ-discounted sum of shaped rewards, we exactly obtain the generalized advantage estimators. As shown previously, λ = 1 gives an unbiased estimate, whereas λ < 1 gives a biased estimate.

5. VALUE FUNTION ESTIMATION

Trust region method to optimize the value function..

When using a nonlinear function approximator to represent the value function,
the simplest approach is to solve a nonlinear regression problem:

$$\underset{\phi}{\text{minimize}} \sum_{n=1}^{N} \|V_\phi(s_n) - \hat{V}_n\|^2, \tag{28}$$

we used a trust region method to optimize the value function in each iteration of a batch optimization procedure. The trust region helps us to avoid overfitting to the most recent batch of data.

$$\begin{aligned} &\underset{\phi}{\text{minimize}} && \sum_{n=1}^{N} \|V_\phi(s_n) - \hat{V}_n\|^2 \\ &\text{subject to} && \frac{1}{N} \sum_{n=1}^{N} \frac{\|V_\phi(s_n) - V_{\phi_{\text{old}}}(s_n)\|^2}{2\sigma^2} \leq \epsilon. \end{aligned} \tag{29}$$

5. VALUE FUNTION ESTIMATION

Trust region method to optimize the value function..

$$\begin{aligned} \underset{\phi}{\text{minimize}} \quad & \sum_{n=1}^{N} \|V_\phi(s_n) - \hat{V}_n\|^2 \\ \text{subject to} \quad & \frac{1}{N}\sum_{n=1}^{N} \frac{\|V_\phi(s_n) - V_{\phi_{\text{old}}}(s_n)\|^2}{2\sigma^2} \le \epsilon. \end{aligned} \tag{29}$$

$$\sigma^2 = \frac{1}{N}\sum_{n=1}^{N} \|V_{\phi_{old}}(s_n) - \hat{V}_n\|^2$$

Above constraint is equivalent to constraining the average KL divergence between the previous and new.

And to compute an approximate solution to the trust region problem,

it is used the conjugate gradient algoritm.

$$\begin{aligned} \underset{\phi}{\text{minimize}} \quad & g^T(\phi - \phi_{\text{old}}) \\ \text{subject to} \quad & \frac{1}{N}\sum_{n=1}^{N} (\phi - \phi_{\text{old}})^T H (\phi - \phi_{\text{old}}) \le \epsilon. \end{aligned} \tag{30}$$

6. EXPERIMENTS

Trust region method to optimize the policy(TRPO)..

TRPO updates the policy by approximately solving the following constrained optimization proble m each iteration:

$$\begin{aligned}
&\underset{\theta}{\text{minimize}}\ L_{\theta_{old}}(\theta) \\
&\text{subject to}\ \overline{D}_{\mathrm{KL}}^{\theta_{old}}(\pi_{\theta_{old}}, \pi_\theta) \le \epsilon \\
&\text{where}\ L_{\theta_{old}}(\theta) = \frac{1}{N}\sum_{n=1}^{N} \frac{\pi_\theta(a_n \mid s_n)}{\pi_{\theta_{old}}(a_n \mid s_n)} \hat{A}_n \\
&\overline{D}_{\mathrm{KL}}^{\theta_{old}}(\pi_{\theta_{old}}, \pi_\theta) = \frac{1}{N}\sum_{n=1}^{N} D_{KL}(\pi_{\theta_{old}}(\cdot \mid s_n) \,\|\, \pi_\theta(\cdot \mid s_n)) \qquad (31)
\end{aligned}$$

As described in (Schulman et al., 2015), we approximately solve this problem by linearizing the o bjective and quadraticizing the constraint.

6. EXPERIMENTS

The whole algorithm for iteratively

updating policy and value function is given below:

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

Note that the policy update θi → θi+1 is performed using the value function Vφi for advantage estimation, not Vφi+1 . Additional bias would have been introduced if we updated the value function first.

6. EXPERIMENTS

Questions:

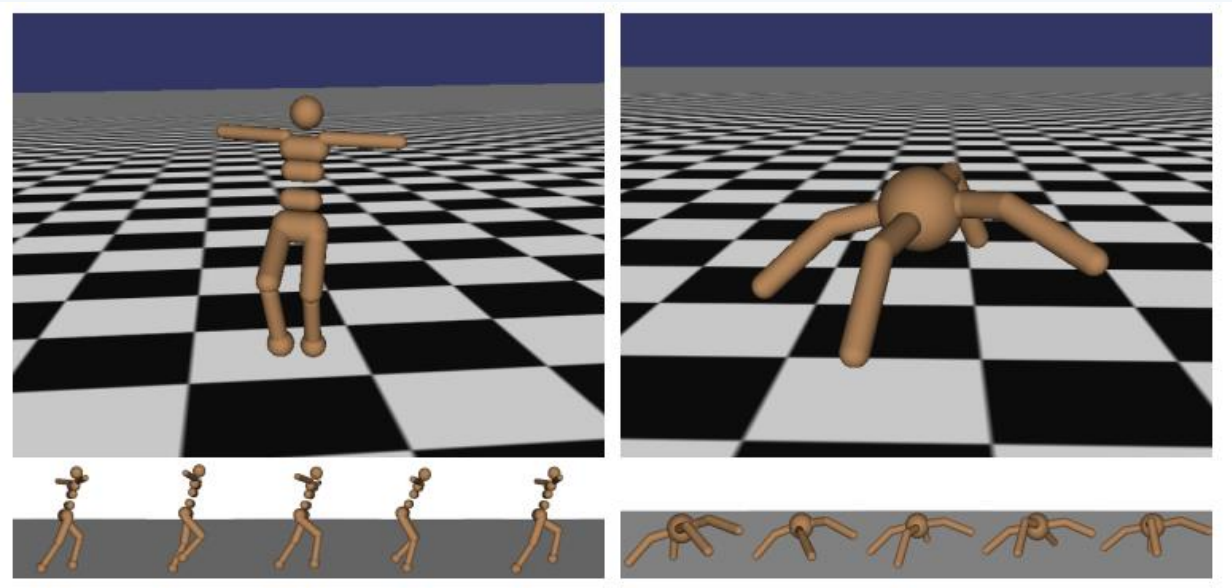
1. What is the empirical effect of varying $\lambda \in [0, 1]$ and $\gamma \in [0, 1]$ when optimizing episodic total reward using *generalized advantage estimation*?

2. Can generalized advantage estimation, along with trust region algorithms for policy and value function optimization, be used to optimize large neural network policies for challenging control problems?

6. EXPERIMENTS

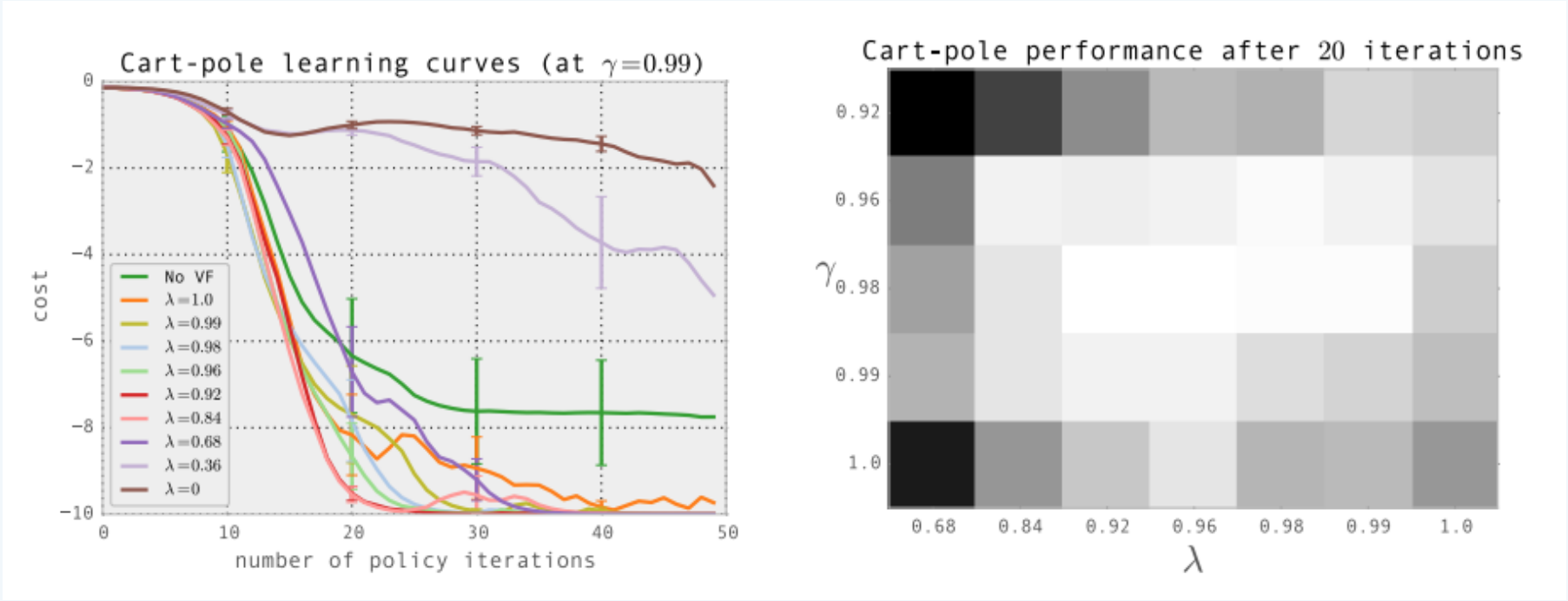
Experimental Setup

1. **Classic cart-pole**
2. Challenging 3D locomotion tasks
   - **bipedal locomotion**
   - **quadrupedal locomotion**
   - **dynamically standing up, for biped**



6. EXPERIMENTS

Task 1. Cart-Pole



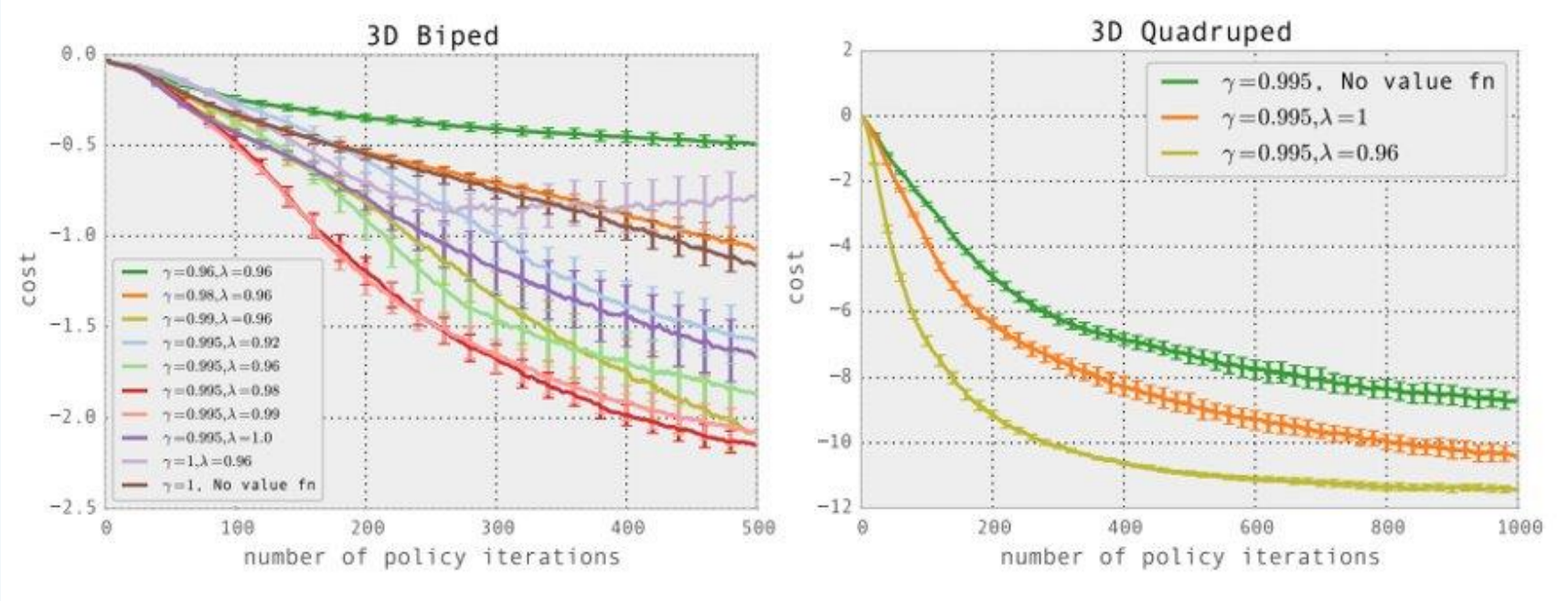
Left: learning curves for cart-pole task, using GAE with varying values of λ at γ = 0.99.
The fastest policy improvement is obtain by intermediate values of λ in the range [0.92, 0.98].

Right: performance after 20 iterations of policy optimization, as γ and λ are varied.
White means higher reward. The best results are obtained at intermediate values of both.

6. EXPERIMENTS

Task 2. 3D Biped &

Task 3. 3D Quadruped



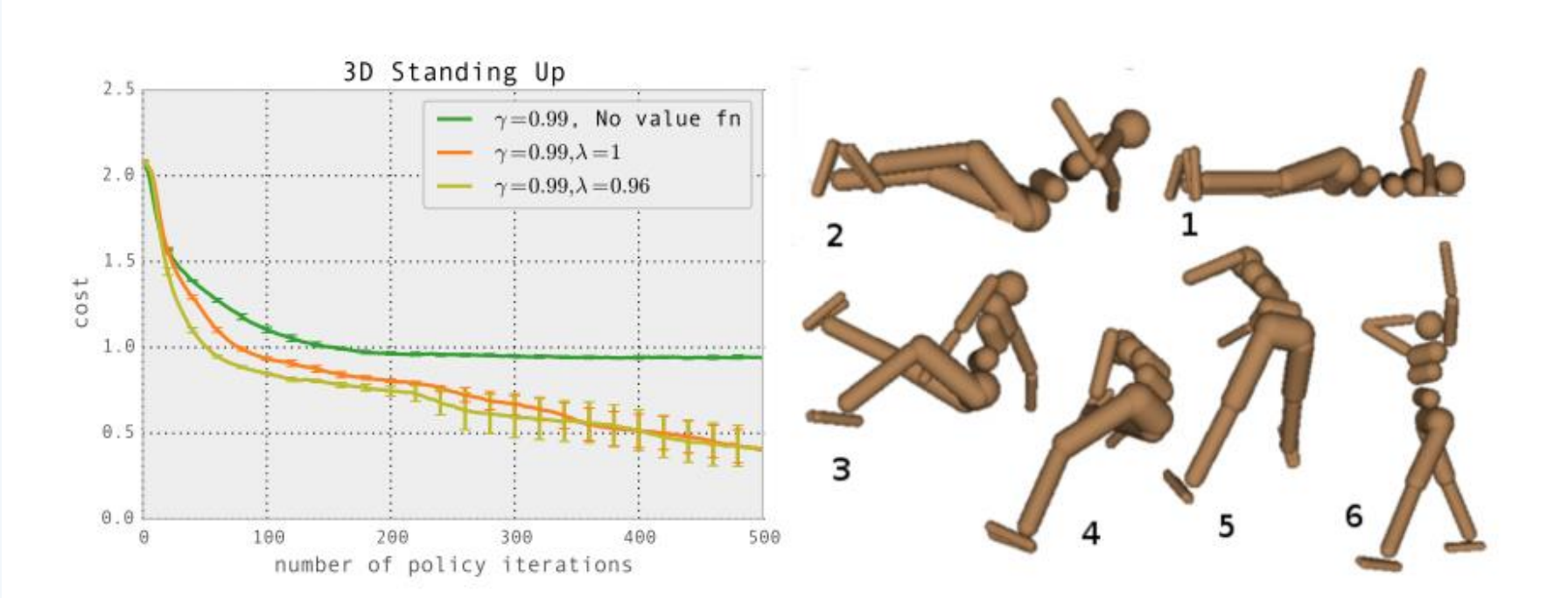
Left: learning curves for 3D bipedal locomotion, averaged across nine runs of the algorithm.

Right: learning curves for 3D quadrupedal locomotion, averaged across five runs

6. EXPERIMENTS

Task 4.

3D Standing up



(a) Learning curve from quadrupedal walking,
(b) learning curve for 3D standing up,
(c) clips from 3D standing up

7. DISCUSSION

- **Challenging Issues**

  policy gradient methods has been limited, largely due to their high sample complexity.

  -> the key to variance reduction is to obtain good estimates of the advantage function.

- **Contributions**

  We provided and justified the generalized advantage estimator(GAE),

  which has **two parameters γ, λ** which adjust the bias-variance tradeoff.

  And combine this idea with TRPO and a trust region algorithm that optimizes a value function,

  so that we are able to learn to solve difficult control tasks that have previously been out of reach fo

  r generic reinforcement learning methods.

7. DISCUSSION

Future Works

- how to adjust the estimator parameters γ, λ in an adaptive or automatic way.

- the relationship between value function estimation error and policy gradient estimation error.

  If this relationship were known, we could choose an error metric for value function fitting that is well-matched to the quantity of interest, which is typically the accuracy of the policy gradient estimation.

- a shared function approximation architecture for the policy and the value function