Action-dependent Control Variates for Policy Optimization via Stein Identity

Hao Liu* ²  Yihao Feng* ¹  Yi Mao ³  Dengyong Zhou ⁴  
Jian Peng⁵  Qiang Liu¹

¹UT Austin  ²UESTC  ³Microsoft  ⁴Google  ⁵UIUC
Reinforcement Learning with Policy Gradients

Goal: learn policy $\pi_\theta(a|s)$ to maximize the expected reward:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\rho_\pi, \pi} \left[ \nabla_\theta \log \pi(a|s) Q^\pi(s, a) \right],$$

$$\approx \frac{1}{n} \sum_{t=1}^{n} \nabla_\theta \log \pi(a_t | s_t) \hat{Q}^\pi(s_t, a_t)$$

Draw $(a_t, s_t)_{t=1}^{n}$ by running policy $\pi$.

Key problem: large variance, sample-intensive.
Control Variates

- Goal: estimate expectation $\mu = \mathbb{E}[f]$, where $f$ has high variance.

- Find a function (control variate) $g(x)$ such that $\mathbb{E}[g] = c$, and then

$$\mu = \mathbb{E}[f] = \mathbb{E}[f - g] + c \approx \frac{1}{n} \sum_{i=1}^{n} [f(x_i) - g(x_i)] + c$$

- Pick $g$ so that $f \approx g + c$, and hence $\text{var}(f - g)$ is small.

- **Key**: Find math identity $\mathbb{E}[g] = c$ that holds for a large class of $g$. 
Identity used in policy gradient:

\[ \mathbb{E}_{\pi(a|s)}[\nabla \theta \log \pi(a|s) \phi(s)] = 0. \]

Policy gradient with control variate:

\[ \nabla J(\theta) = \mathbb{E}_{\pi}[\nabla \theta \log \pi(a|s)(Q^\pi(s, a) - \phi(s))]. \]

**REINFORCE:** \( \phi(s) = b \), constant baseline.

**Advantage Actor-Critic (A2C):** \( \phi(s) = V^\pi(s) \), value function.

**Problem:** \( \phi(s) \) can not depend on action \( a \). Can we design more flexible action-dependent baseline?
Control Variates Using Stein’s Identity

**Idea:** *Stein’s identity:*

\[
\mathbb{E}_{\pi(a|s)} \left[ \nabla_a \log \pi(a|s) \phi(s, a) + \nabla_a \phi(s, a) \right] = 0, \quad \forall s.
\]

- Widely used in statistics and inference, but cannot be directly used for policy gradient because it involves \( \nabla_a \) instead of \( \nabla_\theta \).

- For “reparameterizable” policy \( a = f_\theta(s, \xi) \), Stein’s identity reduces to

\[
\mathbb{E}_{\pi(a|s)} \left[ \nabla_\theta \log \pi(a|s) \phi(s, a) \right] = \mathbb{E}_{\pi(a|s)} \left[ \nabla_\theta f_\theta(s, \xi) \nabla_a \phi(s, a) \right].
\]
“Reparameterizable” policy: \(a = f_\theta(s, \xi)\), Stein’s identity reduces to

\[
\mathbb{E}_{\pi(a|s)} [\nabla_\theta \log \pi(a|s) \phi(s, a)] = \mathbb{E}_{\pi(a|s)} [\nabla_\theta f_\theta(s, \xi) \nabla_a \phi(s, a)].
\]

Related to reparameterization trick (Kingma & Welling 2013):

Define \(L(\theta) = \mathbb{E}_\pi[\phi(s, a)]\)
then there are two ways to write the gradient:

\[
\nabla L(\theta) = \mathbb{E}_{\pi(a|s)} [\nabla_\theta \log \pi(a|s) \phi(s, a)] = \mathbb{E}_{\pi(a|s)} [\nabla_\theta f_\theta(s, \xi) \nabla_a \phi(s, a)].
\]

- log likelihood ratio
- reparameterization trick
The policy gradient can be rewrote into

\[
\nabla_\theta J(\theta) = \mathbb{E}_\pi [\nabla_\theta \log \pi(a|s)(Q^\pi(s, a) - \phi(s, a)) + \nabla_\theta f_\theta(s, \xi)\nabla_a \phi(s, a)].
\n\]

- Generalizes REINFORCE ($\phi = b$) and A2C ($\phi(s, a) = \phi(s)$).
- Q-prop (Gu et.al. 17): $\phi(s, a)$ is a \underline{linear function} of action $a$.
- For Gaussian policies, \underline{Stein’s identity can be applied twice} to further reduce the variance.
Specifying the Parametric Form of $\phi$

- **Linear (motivated by Q-prop):**

$$\phi_w(s, a) = V_w(s) + \langle \nabla_a q_w(a, \mu_\pi(s)), (a - \mu_\pi(s)) \rangle,$$

where $q_w$ is a parametric function designed for estimating the Q function $Q^\pi$.

- **Quadratic:**

$$\phi_w(s, a) = V_w(s) - (a - \mu_w(s))^\top A_w(a - \mu_w(s)).$$

- **MLP.** $\phi_w(s, a)$ is assumed to be a neural network with two hidden layers.
How to Optimize the Parameter $w$ of $\phi$?

- **Method 1:** Minimizing the empirical variance of gradient estimator:

  $\min_w \text{var}(\hat{\nabla}_\theta J(\theta))$

- **Method 2:** Fitting Q function (set $\phi(s, a)$ to be close to $Q^\pi(s, a)$):

  $\min_w \sum_{t=1}^{n} (\phi_w(s_t, a_t) - R_t)^2,$

  where $R_t$ is the estimation of the reward.
Experiments

Comparing the variance of different gradient estimators (on Walker2d-v1)

![Graph comparing variance of different gradient estimators](image)

[Liu et al.](#)
Policy Optimization

Combing our gradient estimator with proximal policy optimization (PPO)

Hypothesis and Experiment

- HumanoidStandup-v1
- Humanoid-v1
- Walker2d-v1
- Ant-v1
- Hopper-v1
- HalfCheetah-v1

Liu et al.
Related Work

   - The same action-dependent control variate formula.
   - Additional experiments on discrete VAE (through Gumbel-Softmax relaxation)

   - Assumption: stochastic policies can be factored:
     \[
     \pi_\theta(a_t|s_t) = \prod_{i=1}^{m} \pi_\theta(a_t^i|s_t).
     \]
   - Conditionally independent actions
     \[
     J(\theta) = \mathbb{E}_{\rho_\pi, \pi} \left[ \sum_{i=1}^{m} \nabla_\theta \log \pi_\theta(a_t^i|s_t)(\hat{Q}(s_t, a_t) - b_i(s_t, a_t^{-i})) \right]
     \]
**Scaling action dimension**

- **Wu et al.** Variance reduction for policy gradient with action-dependent factorized baselines, ICLR 2018.

- The example is a one-step MDP comprising of a single state, \( S = \{0\} \), an \( m \)-dimensional action space, \( A = \mathbb{R}^m \), and a fixed vector \( c \in \mathbb{R}^m \).

  The reward is given as the negative squared \( l_2 \) loss of the action vector, 
  \[ r(s, a) = -\|a - c\|_2^2. \]

