Policy Optimization with Second-Order Advantage Information

Jiajin Li [*1], **Baoxiang Wang**[*1], **Shengyu Zhang**[1,2]

[1] The Chinese University of Hong Kong

[2] Tencent

{jjli,bxwang,syzhang}@cse.cuhk.edu.hk

Abstract

Policy optimization on high-dimensional continuous control tasks exhibits its difficulty caused by the large variance of the policy gradient estimators. We present the action subspace dependent gradient (ASDG) estimator which incorporates the Rao-Blackwell theorem (RB) and Control Variates (CV) into a unified framework to reduce the variance. To invoke RB, our proposed algorithm (POSA) learns the underlying factorization structure among the action space based on the second-order advantage information. POSA captures the quadratic information explicitly and efficiently by utilizing the wide & deep architecture. Empirical studies show that our proposed approach demonstrates the performance improvements on high-dimensional synthetic settings and OpenAI Gym's MuJoCo continuous control tasks.

1 Introduction

Deep reinforcement learning (RL) algorithms have been widely applied in various challenging problems, including video games [Mnih *et al.*, 2015], board games [Silver *et al.*, 2017], robotics [Levine *et al.*, 2016], dynamic routing [Wu *et al.*, 2017; Li *et al.*, 2016], and continuous control tasks [Schulman *et al.*, 2017; Lillicrap *et al.*, 2015]. An important approach among these methods is policy gradient (PG). Since its inception [Williams, 1992], PG has been continuously improved by the Control Variates (CV) [Oates *et al.*, 2017] theory. Examples are REINFORCE [Williams, 1992], Advantage actor-critic (A2C) [Mnih *et al.*, 2016], Q-prop [Gu *et al.*, 2016], and action-dependent baselines [Liu *et al.*, 2018; Grathwohl *et al.*, 2018; Tucker *et al.*, 2018]. However, when dealing with high-dimensional action spaces, CV has limited effects regarding the sample efficiency. Rao-Blackwell theorem (RB) [Casella and Robert, 1996], though not heavily adopted in policy gradient, is commonly used with CV to address high-dimensional spaces [Ranganath *et al.*, 2014].

Motivated by the success of RB in high-dimensional spaces [Ranganath *et al.*, 2014], we incorporate both RB and CV into a unified framework. We present the action subspace dependent gradient (ASDG) estimator. ASDG first breaks the original high dimensional action space into several low dimensional action subspaces and replace the expectation (i.e., policy gradient) with its conditional expectation over subspaces (RB step) to reduce the sample space. A baseline function associated with each of the corresponding action subspaces is used to further reduce the variance (CV step). While ASDG is benefited from both RB and CV's ability to reduce the variance, we show that ASDG is unbiased under relatively weak assumptions over the advantage function.

The major difficulty to invoke RB is to find a satisfying action domain partition. Novel trials such as [Wu *et al.*, 2018] utilize RB under the conditional independence assumption which assumes that the policy distribution is fully factorized with respect to the action. Whilst it dramatically reduces the estimation variance, such a strong assumption limits the policy distribution flexibility and [Wu *et al.*, 2018] is conducting the optimization in a restricted domain. In our works, we show that Hessian of the advantage with respect to the action is theoretically connected with the action space structure. Specifically, the block-diagonal structure of Hessian is corresponding to the partition of the action space. We exploit such second-order information with the evolutionary clustering algorithm [Chakrabarti *et al.*, 2006] to learn the underlying factorization structure in the action space. Instead of the vanilla multilayer perceptron, we utilize the wide & deep architecture [Cheng *et al.*, 2016] to capture such information explicitly and efficiently. With the second-order advantage information, ASDG finds the partition that approximates the underlying structure of the action space.

We evaluate our method on a variety of reinforcement learning tasks, including a high-dimensional synthetic environment and several OpenAI Gym's MuJoCo continuous control environments. We build ASDG and POSA on top of proximal policy optimization (PPO), and demonstrate that ASDG consistently obtains the ideal balance: while improving the sample efficiency introduced by RB [Wu *et al.*, 2018], it keeps the accuracy of the feasible solution [Liu *et al.*, 2018]. In environments where the model assumptions are satisfied or minimally violated empirically, while not trivially satisfied by [Wu *et al.*, 2018], POSA outperforms previous studies with the overall cumulated rewards it achieves. In the continuous control tasks, POSA is either competitive or superior, depend-

---

[*]These authors contribute equally to this work. Correspondence to: Jiajin Li and Baoxiang Wang.

ing on whether the action space exhibits its structure under the environment settings.

2 Background

2.1 Notation

We present the canonical reinforcement learning (RL) formalism in this section. Consider policy learning in the discrete-time Markov decision process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, r, \rho_0, \gamma)$ where $\mathcal{S} \in \mathbb{R}^n$ is the n dimensional state space, $\mathcal{A} \in \mathbb{R}^m$ is the m dimensional action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^+$ is the environment transition probability function, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, ρ_0 is the initial state distribution and $\gamma \in (0, 1]$ is the unnormalized discount factor. RL learns a stochastic policy $\pi_\theta : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_+$, which is parameterized by θ, to maximize the expected cumulative reward

$$J(\theta) = \mathbb{E}_{s \sim \rho_\pi, a \sim \pi}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)].$$

In the above equation, $\rho_\pi(s) = \sum_{t=1}^{\infty} \gamma^{t-1} \mathbb{P}(s_t = s)$ is the discounted state visitation distribution. Define the value function

$$V^\pi(s_t) = \mathbb{E}_\pi[\sum_{t' \geq t}^{\infty} \gamma^{t'-t} r(s_{t'}, a_{t'})|s_t, \pi]$$

to be the expected return of policy π at state s_t. Define the state-action function

$$Q^\pi(s_t, a_t) = \mathbb{E}_\pi[\sum_{t' \geq t}^{\infty} \gamma^{t'-t} r(s_{t'}, a_{t'})|s_t, a_t, \pi]$$

to be the expected return by policy π after taking the action a_t at the state s_t. We use $\hat{Q}^\pi(s_t, a_t)$ and $\hat{V}^\pi(s_t)$ to denote the empirical function approximator of $Q^\pi(s_t, a_t)$ and $V^\pi(s_t)$, respectively. Define the advantage function to be the gap between the value function and the action-value, as $A^\pi(s_t, a_t) = Q^\pi(s_t, a_t) - V^\pi(s_t)$. To simplify the notation, we focus on the time-independent formulation $J(\theta) = \mathbb{E}_{\pi, \rho_\pi}[r(s, a)]$. According to the policy gradient theorem [Williams, 1992], the gradient of the expected cumulative reward can be estimated as

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

2.2 Variance Reduction Methods

In practice, the vanilla policy gradient estimator is commonly estimated using Monte Carlo samples. A significant obstacle to the estimator is the sample efficiency. We review three prevailing variance reduction techniques in Monte Carlo estimation methods, including Control Variates, Rao-Blackwellization, and Reparameterization Trick.

Control Variates

Consider the case we estimate the expectation $\mathbb{E}_{p(x)}[h(x)]$ with Monte Carlo samples $\{x_i\}_{i=1}^B$ from the underlying distribution $p(x)$. Usually, the original Monte Carlo estimator has high variance, and the main idea of Control Variates is to find the proper baseline function $g(x)$ to partially cancel

out the variance. A baseline function $g(x)$ with its known expectation over the distribution $p(x)$ is used to construct a new estimator $\hat{h}(x) = h(x) - \eta(g(x) - \mathbb{E}_p[g(x)])$, where η is a constant determined by the empirical Monte Carlo samples. The Control Variates method is unbiased but with a smaller variance $Var(\hat{h}(x)) \leq Var(h(x))$ at the optimal value $\eta^* = \frac{Cov(h,g)}{Var(g)}$.

Rao-Blackwellization

Though most of the recent policy gradient studies reduce the variance by Control Variates, the Rao-Blackwell theorem [Casella and Robert, 1996] decreases the variance significantly more than CV do, especially in high-dimensional spaces [Ranganath et al., 2014]. The motivation behind RB is to replace the expectation with its conditional expectation over a subset of random variables. In this way, RB transforms the original high-dimensional integration computation problem into estimating the conditional expectation on several low-dimensional subspaces separately.

Consider a simple setting with two random variable sets $\mathcal{A}$ and $\mathcal{B}$ and the objective is to compute the expectation $\mathbb{E}[h(\mathcal{A}, \mathcal{B})]$. Denote that the conditional expectation $\hat{\mathcal{B}}$ as $\hat{\mathcal{B}} = \mathbb{E}[h(\mathcal{A}, \mathcal{B})|\mathcal{A}]$. The variance inequality $Var(\hat{\mathcal{B}}) \leq Var(h(\mathcal{A}, \mathcal{B}))$ holds as shown in the Rao-blackwell theorem. In practical, when $\mathcal{A}$ and $\mathcal{B}$ are in high dimensional spaces, the conditioning is very useful and it reduces the variance significantly. The case of multiple random variables is hosted in a similar way.

Reparameterization Trick

One of the recent advances in variance reduction is the reparameterization trick. It provides an estimator with lower empirical variance compared with the score function based estimators, as demonstrated in [Kingma and Welling, 2013; Ranganath et al., 2016]. Using the same notation as is in the Control Variates section, we assume that the random variable x is reparameterized by $x = f(\theta, \xi), \xi \sim q(\xi)$, where $q(\xi)$ is the base distribution (e.g., the standard normal distribution or the uniform distribution). Under this assumption, the gradient of the expectation $\mathbb{E}_{p(x)}[h(x)]$ can be written as two identical forms i.e., the score function based form and reparameterization trick based form

$$\mathbb{E}_p[\nabla_\theta \log p(x) h(x)] = \mathbb{E}_q[\nabla_\theta f(\theta, \xi) \nabla_x h(x)]. \quad (1)$$

The reparameterization trick based estimator (the right-hand side term) has relatively lower variance. Intuitively, the reparameterization trick provides more informative gradients by exposing the dependency of the random variable x on the parameter θ.

2.3 Policy Gradient Methods

Previous attempts to reduce the variance mainly focus on the Control Variates method in the policy gradient framework (i.e., REINFORCE, A2C, Q-prop). A proper choice of the baseline function is vital to reduce the variance. The vanilla policy gradient estimator, REINFORCE [Williams, 1992], subtracts the constant baseline from the action-value function,

$$\nabla_\theta J(\theta)_{RF} = \mathbb{E}_\pi[\nabla_\theta \log \pi(a|s)(Q^\pi(s, a) - b)].$$

The estimator in REINFORCE is unbiased. The key point to conclude the unbiasedness is that the constant baseline function has a zero expectation with the score function. Motivated by this, the baseline function is set to be the value function $V^\pi(s)$ in the advantage actor-critic (A2C) method [Mnih *et al.*, 2016], as the value function can also be regarded as a constant under the policy distribution $\pi(a|s)$ with respect to the action a. Thus the A2C gradient estimator is

$$\nabla_\theta J(\theta)_{A2C} = \mathbb{E}_\pi[\nabla_\theta \log \pi(a|s)(Q^\pi(s,a) - V^\pi(s))]$$
$$= \mathbb{E}_\pi[\nabla_\theta \log \pi(a|s)A^\pi(s,a)].$$

To further reduce the gradient estimate variance to acquire a zero-asymptotic variance estimator, [Liu *et al.*, 2018] and [Grathwohl *et al.*, 2018] propose a general action dependent baseline function $b(s,a)$ based on the identity (1). Note that the stochastic policy distribution $\pi_\theta(a|s)$ is reparametrized as $a = f(\theta, s, \xi), \xi \sim q(\xi)$, we rewrite Eq.(1) to get a zero-expectation baseline function as below

$$\mathbb{E}[\nabla_\theta \log \pi(a|s)b(s,a) - \nabla_\theta f(\theta,s,\xi)\nabla_a b(s,a)] = 0. \quad (2)$$

Incorporating with the zero-expectation baseline (2), the general action dependent baseline (GADB) estimator is formulated as

$$\nabla_\theta J(\theta)_{GADB} = \mathbb{E}_\pi[\nabla_\theta \log \pi(a|s)(Q^\pi(s,a) - b(s,a))$$
$$+ \nabla_\theta f(\theta,s,\xi)\nabla_a b(s,a)]. \quad (3)$$

3 Methods

3.1 Construct the ASDG Estimator

We present our action subspace dependent gradient (ASDG) estimator by applying RB on top of the GADB estimator. Starting with Eq.(3), we rewrite the baseline function in the form of $b(s,a) = V^\pi(s) + c(s,a)$. The GADB estimator in Eq.(3) is then formulated as

$$\nabla_\theta J(\theta)_{GADB} = \mathbb{E}_\pi[\nabla_\theta \log \pi(a|s)(A^\pi(s,a) - c(s,a))$$
$$+ \nabla_\theta f(s,\theta,\xi)\nabla_a c(s,a)].$$

Assumption 1 (Advantage Quadratic Approximation)
Assume that the advantage function $A^\pi(s,a)$ can be locally second-order Taylor expanded with respect to a at some point a^*, that is,

$$A^\pi(a,s) \approx A^\pi(a^*,s) + \nabla_a A^\pi(a,s)|_{a=a^*}^T(a-a^*)$$
$$+ \frac{1}{2}(a-a^*)^T \nabla_{aa}A^\pi(a,s)|_{a=a^*}(a-a^*) \quad (4)$$

The baseline function $c(s,a)$ is chosen from the same family.

Assumption 2 (Block Diagonal Assumption) *Assume that the row-switching transform of Hessian $\nabla_{aa}A^\pi(a,s)|_{a=a^*}$ is the block diagonal matrix $diag(M_1,...,M_k)$, where $\sum_{k=1}^K dim(M_k) = m$.*

Based on the assumption (1) and (2), the advantage function $A^\pi(s,a)$ can be divided into K independent subspaces,

$$A^\pi(s,a) = \sum_{k=1}^K A_k^\pi(s, a_{(k)}),$$

where $a_{(k)}$ denotes the projection of action to the k-th action subspace with respect to M_k. The baseline function $c(s,a)$ is divided in the same way.

Theorem 3 (ASDG Estimator) *If the advantage function $A^\pi(s,a)$ and the baseline function $c(s,a)$ satisfy assumption (1) and (2), the ASDG estimator $\nabla_\theta J(\theta)_{ASDG}$ is*

$$\sum_{k=1}^K \mathbb{E}_{\pi(a_{(k)}|s)}[\nabla_\theta \log \pi(a_{(k)}|s)(A^\pi(s, a_{(k)})$$
$$- c(s,(a_{(k)}, \tilde{a}_{(-k)}))) - \nabla_\theta f_k(\theta, s, \xi)\nabla_{a_{(k)}} c_k(s, a_{(k)})],$$

where $\nabla_\theta f(\theta, s, \xi) \in \mathbb{R}^{N_\theta \times m}$ is divided into K parts as $\nabla_\theta f = [\nabla_\theta f_1, ..., \nabla_\theta f_K]$ and N_θ is the dimension of θ.

Proof 3.1 *Using the fact that*

$$\mathbb{E}_{\pi(a|s)}[.] = \mathbb{E}_{\pi(a_{(k)}|s)}\mathbb{E}_{\pi(a_{(-k)}|a_{(k)},s)}[.]$$

where $a_{(-k)}$ represents the dimensions within a that are complementary to $a_{(k)}$. Denote that $\mathbb{E}_{\pi(a|s)}[.] = \mathbb{E}_{\pi_k}\mathbb{E}_{\pi_{-k}}$ to simplify the notation. With the assumptions (1) and (2), $\nabla J(\theta)_{ASDG}$ can be derived as,

$$= \mathbb{E}_{\pi_k}\mathbb{E}_{\pi_{-k}}[(\nabla_\theta \log \pi(a_{(k)}|s) + \nabla_\theta \log \pi(a_{(-k)}|a_{(k)},s))$$
$$(A_k^\pi(s, a_{(k)}) + \sum_{i \neq k} A_i^\pi(s, a_{(i)}) - c_k(s, a_{(k)})$$
$$- \sum_{i \neq k} c_i(s, a_{(i)})) + \sum_{k=1}^K \nabla_\theta f_k(s, a_{(k)})\nabla_{a_{(k)}} c_k(s, a_{(k)})]$$
$$= \mathbb{E}_{\pi_k}\mathbb{E}_{\pi_{-k}}[\nabla_\theta \log \pi(a_{(k)}|s)(A_k^\pi - c_k) - \nabla_\theta f_k \nabla_{a_{(k)}} c_k]$$
$$+ \mathbb{E}_{\pi_k}\mathbb{E}_{\pi_{-k}}[\nabla_\theta \log \pi(a_{(k)}|s)(\sum_{i \neq k} A_i^\pi - \sum_{i \neq k} c_i)] \quad (5)$$
$$+ \mathbb{E}_{\pi_k}\mathbb{E}_{\pi_{-k}}[\nabla_\theta \log \pi(a_{(-k)}|a_{(k)},s)(A_k^\pi - c_k)] \quad (6)$$
$$+ \mathbb{E}_{\pi_k}\mathbb{E}_{\pi_{-k}}[\nabla_\theta \log \pi(a_{(-k)}|a_{(k)},s)((\sum_{i \neq k} A_i^\pi - \sum_{i \neq k} c_i))$$
$$- \sum_{i \neq k} \nabla_\theta f_i \nabla_{a_{(i)}} c_i]$$
$$\overset{(\clubsuit)}{=} \mathbb{E}_{\pi_k}[\nabla_\theta \log \pi(a_{(k)}|s)(A_k^\pi - c_k) - \nabla_\theta f_k \nabla_{a_{(k)}} c_k]$$
$$+ \mathbb{E}_{\pi_{-k}}[\nabla_\theta \log \pi(a_{(-k)}|a_{(k)},s)((\sum_{i \neq k} A_i^\pi - \sum_{i \neq k} c_i))$$
$$- \sum_{i \neq k} \nabla_\theta f_i \nabla_{a_{(i)}} c_i] \quad (7)$$
$$\overset{(\heartsuit)}{=} \sum_{k=1}^K \mathbb{E}_{\pi_k}[\nabla_\theta \log \pi(a_{(k)}|s)(A_k^\pi - c_k) - \nabla_\theta f_k \nabla_{a_{(k)}} c_k]$$
$$= \sum_{k=1}^K \mathbb{E}_{\pi_k}[\nabla_\theta \log \pi(a_{(k)}|s)(A_k^\pi + \sum_{i \neq k} A_i^\pi - c_k - \sum_{i \neq k} c_i)$$
$$- \nabla_\theta f_k \nabla_{a_{(k)}} c_k]$$
$$= \sum_{k=1}^K \mathbb{E}_{\pi_k}[\nabla_\theta \log \pi(a_{(k)}|s)(A^\pi(s,a) - c(s, a_{(k)}, \tilde{a}_{(-k)}))$$
$$- \nabla_\theta f_k \nabla_{a_{(k)}} c_k] \quad (8)$$

where ($\clubsuit$) holds as term (5) and term (6) equal to zero (i.e., actually the property that the expectation of the score function is zero) and ($\heartsuit$) is expanded by induction. ∎

Our assumptions are relatively weak compared with previous studies on variance reduction for policy optimization. Different from the strong fully factorization policy distribution assumed in [Wu *et al.*, 2018], our method relaxes this assumption to the constraints on the advantage function $A^\pi(s, a)$ with respect to the action space instead. Similar to that, we just use this assumption to obtain the structured factorization action subspaces to invoke the Rao-Blackwellization and our estimator does not introduce additional bias.

Connection with other works - When we assume that the Hessian matrix of the advantage function has no block diagonal structure under any row switching transformation (i.e., $K = 1$), ASDG in Theorem. 3 is the one inducted in [Liu *et al.*, 2018] and [Grathwohl *et al.*, 2018]. Otherwise, if we assume that Hessian is diagonal (i.e., $K = m$), the baseline function $c(s, a_{(k)}, \tilde{a}_{(-k)})$ equals to $\sum_{i \neq k} c_i(s, a_{(i)})$, which means that each action dimension is independent with its baseline function. Thus, the estimator in [Wu *et al.*, 2018] is obtained.

Selection of the baseline functions $c(s, a)$ - Two approaches exist to find the baseline function, including minimizing the variance of the PG estimator or minimizing the square error between the advantage function and the baseline function [Liu *et al.*, 2018; Grathwohl *et al.*, 2018]. Minimizing the variance is hard to implement in general, as it involves the gradient of the score function with respect to the baseline function parameter. In our work, we use a neural network advantage approximation as our baseline function by minimizing the square error. Under the assumption that the variance of reparametrization term $\nabla_\theta f_k(\theta, s, \xi) \nabla_{a_{(k)}} c_k(s, a_{(k)})$ is closed to zero, the two methods yield the same result.

3.2 Action Domain Partition based on Wide & Deep Advantage Net

Temporal Difference (TD) learning methods such as Generalized Advantage Estimation (GAE) [Degris *et al.*, 2012; Schulman *et al.*, 2015] allows us to obtain the estimation $\hat{A}(s, a)$ based on the value function $V^w(s)$ via

$$\hat{A}(s_t, a_t) = \sum_{t' \geq t}^{T} (\lambda\gamma)^{t'-t} \delta_{t'}, \tag{9}$$

where

$$\delta_t = \mathbb{E}[r_t + \gamma V^w(s_{t+1}) - V^w(s_t)] \tag{10}$$

and λ is the factor in GAE. GAE further reduces the variance and avoid the action gap at the cost of a small bias.

As shown in the assumption (2), we want to use the block diagonal matrix to approximate the true Hessian matrix for acquiring the structure information in the action space. Nevertheless, we cannot obtain the second-order information $\nabla_{aa} A^w(s, a)$ with the advantage realization value in GAE identity (9). In this paper, we ease the above issue by designing a new neural network approximation $A^\mu(s, a)$ to smoothly interpolate several realization values $\hat{A}(s, a)$, which equals to minimize the square error,

$$\min_\mu ||\hat{A}(s, a) - A^\mu(s, a)||^2. \tag{11}$$

In the above advantage approximation example, the Hessian computation is done by first approximating the original advantage realization value, then differentiating this advantage approximator to obtain an approximate Hessian. However, for any finite number of data points there exists an infinite number of functions, with arbitrarily satisfied Hessians and gradients, which can perfectly approximate the advantage realization values. Optimizing such approximations leads to unstable training and is prone to yield poor results. To alleviate this issue, we propose a novel wide & deep architecture based advantage net. In this way, we divide the advantage approximator into two parts, including the quadratic term and the DNN component

$$A^\mu(s, a) = \beta_1 * A_{wide} + \beta_2 * A_{DNN},$$

where β_1 and β_2 are the importance weights. Subsequently, we make use of Factorization Machine (FM) model as our wide components

$$A_{wide}(s, a) = w_0(s) + w_1(s)^T a + w_2(s) w_2(s)^T \odot aa^T,$$

where $w_0(s) \in \mathbb{R}^1$, $w_1(s) \in \mathbb{R}^{m \times 1}$ and $w_2(s) \in \mathbb{R}^{m \times n_{fm}}$ are the coefficients associated with the action. Also, n_{fm} is the dimension of latent feature space in the FM model. Note that $A \odot B = \sum_i \sum_j A_{ij} B_{ij}$.

To increase the signal-to-noise ratio of the second-order information, we make use of wide components Hessian $w_2(s) w_2(s)^T$ as our Hessian approximator in our work. On the one hand, we can compute it via the forward propagation with low computational costs. On the other hand, the Deep component involves large noise and uncertainties and we obtain stable and robust Hessian by ruling out the deep component.

The Hessian matrix contains both positive and negative values. However, we concern only the correlation of pairwise action dimensions, which can be directly represented by the absolute value of Hessian. For instance, consider a quadratic function $f(x) = a + b^T x + x^T C x, x \in \mathbb{R}^m$, it can also be rewritten as $f(x) = a + \sum_{i=1}^{m} b_i x_i + \sum_{i=1}^{m} \sum_{j=1}^{m} C_{ij} x_i x_j$. Therefore, any entry in the Hessian matrix satisfies $\frac{\partial^2 f(x)}{\partial x_i \partial x_j} = C_{ij}$. When C_{ij} is close to zero, x_i and x_j is nearly independent. Thus we can decompose the function $f(x)$ as two terms and optimize these separately.

In this work, we modify the evolutionary clustering algorithm in [Chakrabarti *et al.*, 2006] by using the absolute approximating Hessian $|w_2(s) w_2(s)^T|$ as the affinity matrix in the clustering task. In other words, each row in the absolute Hessian is regarded as a feature vector of that action dimension when running the clustering algorithm.

At last, the final practical algorithm POSA (1) for action subspace dependent gradient (ASDG) estimator is as follows. In the theorem (3), it involves two advantage information approximation terms to construct the ASDG estimator. Specifically, we utilize the generalized advantage estimation (9) as our first approximation $A^\pi(s, a)$ and use another wide & deep neural network based advantage approximator $A^\mu(s, a)$ to be our baseline function $c(s, a)$ in our algorithm POSA.

Algorithm 1: POSA

Input: number of iterations N, batch size B, number of
      subspaces K, initial policy parameter θ, initial
      value and advantage parameters w, μ;

Output: Policy optimal parameter θ

for *each iteration n in $[N]$* **do**
    Collect a batch of trajectory data $\{s_t^{(i)}, a_t^{(i)}, r_t^{(i)}\}_{i=1}^{B}$;
    for *M_θ iterations* **do**
        Update θ by one SGD step using PPO with
        action subspace estimator in Theorem (3);
    end
    for *M_w iterations* **do**
        Update w and μ by minimizing $||V^w(s_t) - R_t||_2^2$,
        $||\hat{A}(s_t, a_t) - A^\mu(s_t, a_t)||_2^2$ in one SGD step ;
    end
    Estimate $\hat{A}(s_t, a_t)$ using $V^w(s_t)$ by GAE (9);
    Calculate the action subspace partition based on the
      absolute Hessian $|w_2(s)w_2(s)^T|$ by the
      evolutionary clustering algorithm;
end

---

4 Experiments and Results

We demonstrate the sample efficiency and accuracy of ASDG and Alg.(1) in terms of both performance and variance. ASDG is compared with several of the state-of-the-art gradient estimators.

- **Action dependent factorized baselines (ADFB)** [Wu *et al.*, 2018] assumes fully factorized policy distributions, and uses $A(s, (\bar{a}_{(k)}, a_{(-k)}))$ as the k-th dimensional baseline. The subspace $a_{(k)}$ is restricted to contain only one dimension, which is the special case of ASDG with $K = m$.

- **Generalized advantage dependent baselines (GADB)** [Liu *et al.*, 2018; Grathwohl *et al.*, 2018] uses a general baseline function $c(s, a)$ which depends on the action. It does not utilize Rao-Blackwellization and is our special case when $K = 1$.

4.1 Implementation Details

Our algorithm is built on top of PPO where the advantage realization value is estimated by GAE. Our code is available at https://github.com/wangbx66/Action-Subspace-Dependent. We have a policy network for PPO and a value network used by GAE that has the same architecture as in [Mnih *et al.*, 2016; Schulman *et al.*, 2017]. We utilize a third network which estimates the advantage $A^\mu(s, a)$ smoothly by solving Eq.(11) to be our baseline function $c(s, a)$. The network computes the advantage and the Hessian matrix approximator $w_2(s)w_2(s)^T$ by forward propagation. It uses the wide & deep architecture: For the wide component, the state is mapped to $w_1(s)$ and $w_2(s)$ through two-layer MLPs, both with size 128 and $\tanh(\cdot)$ activation. A_{DNN} is a three-layer MLP with size 128 and $\tanh(\cdot)$ activation. Our other parameters are consistent with those in [Schulman *et al.*, 2017] except that we reduce the learning rate (i.e., 3×10^{-4}) by ten times for more stable comparisons.

4.2 Synthetic High-Dimensional Action Spaces

We design a synthetic environment with a wide range of action space dimensions and explicit action subspace structure to test the performance of Alg.(1) and compare that with previous studies. The environment is a one-step MDP where the reward $r(s, a) = \sum_{k=1}^{K} a_{(k)}^T M_k a_{(k)} + \epsilon$ does not depend on the state s (e.g., ϵ is a random noise). In the environment, the action is partitioned into K independent subspaces with a stationary Hessian of the advantage function. Each of the subspace can be regarded as an individual agent. The environment setting satisfies both Assumption (1) and (2).

Fig.1 shows the results on the synthetic environment for ASDG with different dimensions m and number of subspaces K. The legend *ASDG_K* stands for our ASDG estimator with K blocks assumption. For environments with relatively low dimensions such as (a) and (b), all of the algorithms convergences to the same point because of the simplicity of the settings. Both ASDG and ADFB (that incorporates RB) outperforms GADB significantly in terms of sample efficiency while ADFB is marginally better ASDG. For high dimensional settings such as (c) and (d), both ASDG and GADB converge to the same point with high accuracy. Meanwhile, ASDG achieves the convergence significantly faster because of its efficiency. ADFB, though having better efficiency, fails to achieve the competitive accuracy.

We observe an ideal balance between accuracy and efficiency. On the one hand, ASDG trades marginal accuracy for efficiency when efficiency is the bottleneck of the training, as is in (a) and (b). On the other hand, ASDG trades marginal efficiency for accuracy when accuracy is relatively hard to achieve, as is in (c) and (d). ASDG's tradeoff results in the combination of both the merits of its extreme cases.

We also demonstrate that the performance is robust to the assumed K value in (a) when accuracy is not the difficulty. As shown in (a), the performance of ASDG is only decided by its sample efficiency, which is monotonically increased with K. However in complicated environments, an improper selection of K results in the loss of accuracy. Hence, in general, ASDG performs best overall when the K value is set to the right value instead of the maximum.

4.3 OpenAI Gym's MuJoCo Environments

We present the results of the proposed action subspace dependent gradient (ASDG) estimator on common benchmark tasks. These tasks and experiment settings have been widely studied in the deep reinforcement learning community [Duan *et al.*, 2016; Gu *et al.*, 2016; Wu *et al.*, 2018; Liu *et al.*, 2018]. We test ASDG on several environments with relatively high action dimensions, namely Walker2d, Hopper, HalfCheetah, and Ant, shown in Fig.2 and Fig.3. In general, ASDG outperforms ADFB and GADB consistently but performs extraordinarily well when the action space satisfies the dependency assumption (i.e., black diagonal assumption for the advantage function) such as in HalfCheetah.

To investigate the choice of K, we test all the possible K values in Walker2d. The optimal K value is supposed to be between its extreme $K = 1$ and $K = m$ cases. Empirically, we find it effective to conduct a grid search. We consider

(a) Dim=4, K=2

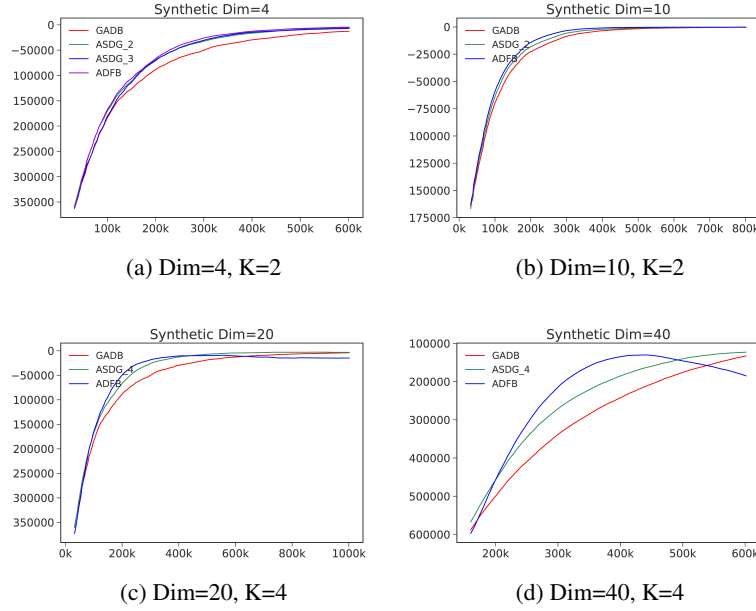(b) Dim=10, K=2

(c) Dim=20, K=4

(d) Dim=40, K=4

Figure 1: Learning curve for synthetic high-dimensional continuous control tasks, varying from 4 to 40 dimensions. At high dimensions, our ASDG estimator provides an ideal balance between the accuracy (i.e., GADB) and efficiency (i.e., ADFB).
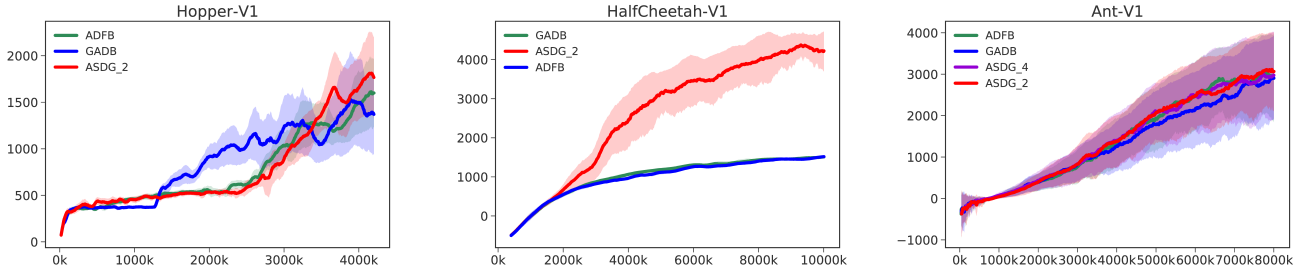

Figure 2: Comparison between two baselines (ADFB, GADB) and our ASDG estimator on various OpenAI Gym Mujoco continuous control tasks, including Hopper-V1 (Dim=3), HalfCheetah-V1 (Dim=6) and Ant-V1 (Dim=8). Our ASDG estimator performs consistently the best across all these tasks.

the automatically approach to finding the optimal K value an interesting future extension.

Figure 3: The choices of action subspace number K in the Walker2d-V1 environment.

5 Conclusion

We propose action subspace dependent gradient (ASDG) estimator, which combines Rao-Blackwell theorem and Control Variates theory into a unified framework to cope with the high dimensional action space. It captures the second-order information of the advantage function via the Wide & Deep architecture and exploits the information to find the dependency structure of the action space. The POSA algorithm reduces variance from the original policy gradient estimator which keeps it unbiased under weaker assumptions than previous studies [Wu *et al.*, 2018]. Our method performs well on a variety of environments including high-dimensional synthetic environment and OpenAI Gym's MuJoCo continuous control tasks. It ideally balances the two extreme cases and demonstrates the merit of both the methods.
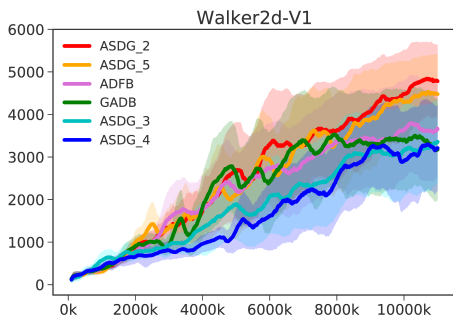
References

[Casella and Robert, 1996] George Casella and Christian P Robert. Rao-blackwellisation of sampling schemes. *Biometrika*, 83(1):81–94, 1996.

[Chakrabarti *et al.*, 2006] Deepayan Chakrabarti, Ravi Kumar, and Andrew Tomkins. Evolutionary clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 554–560. ACM, 2006.

[Cheng *et al.*, 2016] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pages 7–10. ACM, 2016.

[Degris *et al.*, 2012] Thomas Degris, Martha White, and Richard S Sutton. Off-policy actor-critic. *arXiv preprint arXiv:1205.4839*, 2012.

[Duan *et al.*, 2016] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning*, pages 1329–1338, 2016.

[Grathwohl *et al.*, 2018] Will Grathwohl, Dami Choi, Yuhuai Wu, Geoff Roeder, and David Duvenaud. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. In *International Conference on Learning Representations*, 2018.

[Gu *et al.*, 2016] Shixiang Gu, Timothy Lillicrap, Zoubin Ghahramani, Richard E Turner, and Sergey Levine. Qprop: Sample-efficient policy gradient with an off-policy critic. *arXiv preprint arXiv:1611.02247*, 2016.

[Kingma and Welling, 2013] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[Levine *et al.*, 2016] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

[Li *et al.*, 2016] Shuai Li, Baoxiang Wang, Shengyu Zhang, and Wei Chen. Contextual combinatorial cascading bandits. In *International Conference on Machine Learning*, pages 1245–1253, 2016.

[Lillicrap *et al.*, 2015] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[Liu *et al.*, 2018] Hao Liu, Yihao Feng, Yi Mao, Dengyong Zhou, Jian Peng, and Qiang Liu. Action-dependent control variates for policy optimization via stein identity. In *International Conference on Learning Representations*, 2018.

[Mnih *et al.*, 2015] Volodymyr Mnih, KoPieterray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[Mnih *et al.*, 2016] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy P Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, 2016.

[Oates *et al.*, 2017] Chris J Oates, Mark Girolami, and Nicolas Chopin. Control functionals for monte carlo integration. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(3):695–718, 2017.

[Ranganath *et al.*, 2014] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822, 2014.

[Ranganath *et al.*, 2016] Rajesh Ranganath, Dustin Tran, and David Blei. Hierarchical variational models. In *International Conference on Machine Learning*, pages 324–333, 2016.

[Schulman *et al.*, 2015] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Silver *et al.*, 2017] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

[Tucker *et al.*, 2018] George Tucker, Surya Bhupatiraju, Shixiang Gu, Richard E Turner, Zoubin Ghahramani, and Sergey Levine. The mirage of action-dependent baselines in reinforcement learning. *arXiv preprint arXiv:1802.10031*, 2018.

[Williams, 1992] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[Wu *et al.*, 2017] Cathy Wu, Kanaad Parvate, Nishant Kheterpal, Leah Dickstein, Ankur Mehta, Eugene Vinitsky, and Alexandre Bayen. Framework for control and deep reinforcement learning in traffic. In *Intelligent Transportation Systems (ITSC), 2017 IEEE 20th International Conference on*, pages 1–8. IEEE, 2017.

[Wu *et al.*, 2018] Cathy Wu, Aravind Rajeswaran, Yan Duan, Vikash Kumar, Alexandre M.Bayen, Sham Kakade, Igor Mordatch, and Pieter Abbeel. Variance reduction for policy gradient with action-dependent factorized baselines. In *International Conference on Learning Representations*, 2018.