Is Standard Deviation the New Standard? Revisiting the Critic in Deep Policy Gradients

Anonymous Author(s)

Abstract

Policy gradient algorithms have proven to be successful in diverse decision making and control tasks. Although promising, these methods still suffer profoundly from a high sample complexity and instability issues. In this paper, we address these challenges by providing a different approach for training the critic in the actor-critic framework. Namely, we propose a new cost function for the critic: the variance of the value function residual errors instead of the usual mean squared error. Our work, Actor with Variance Estimated Critic (AVEC), builds on recent studies indicating that traditional actor-critic algorithms do not succeed in fitting the true value function, calling for the need to identify a better objective for the critic. In AVEC, the critic learns a better approximation of the relative value of the states/state-action pairs rather than their absolute value as in conventional actor-critic. We prove the theoretical consistency of the new gradient estimator and observe dramatic empirical improvement across a variety of continuous control tasks and algorithms. Furthermore, we validate our method in tasks with sparse rewards, where we provide experimental evidence and theoretical insights.

1 Introduction

Model-free deep reinforcement learning (RL) has been successfully used in a wide range of problem domains, ranging from teaching computers to control robots to playing sophisticated strategy games (Silver et al., 2014; Schulman et al., 2016; Lillicrap et al., 2016; Mnih et al., 2016). State-of-the-art policy gradient algorithms currently combine ingenious learning schemes with neural networks as function approximators in the so-called actor-critic framework (Sutton et al., 2000; Schulman et al., 2017; Haarnoja et al., 2018). While such methods demonstrate great performance on continuous control tasks, many discrepancies persist between what motivates the conceptual framework of these algorithms and what is implemented in practice to obtain maximum gains.

For instance, works on variance reduction often restrict the class of function approximators through different assumptions, then propose a baseline formulation that allows optimal reduction of the variance of the gradient. The value function (Sutton et al., 2000) is the standard baseline used, and recently state-action-dependent baselines have also been proposed (Gu et al., 2016; Liu et al., 2018; Wu et al., 2018). However, new studies (Tucker et al., 2018; Ilyas et al., 2020) indicate that state-of-the-art policy gradient methods (Schulman et al., 2015, 2017) using state-dependent baselines fail to fit the true value function and that the recently proposed state-action-dependent baselines (Gu et al., 2016; Liu et al., 2018; Wu et al., 2018) do not reduce the variance of gradient estimates more than state-dependent ones.

These findings leave the reader somewhat skeptical about actor-critic algorithms, suggesting that baselines are used to improve empirical performance by introducing a bias rather than stabilizing the gradient. The ubiquity of these deep policy gradient challenges indicates the root of the problem: variance-reduction methods do not reduce the variance. We believe that our efforts should not be focused on finding a better baseline, since critics would not be able to fit it anyway (Ilyas et al., 2020). Tucker et al. (2018) argue that "much larger gains could be achieved by instead improving the accuracy of the value function". Following this line of thought, we are interested in how to better approximate the true value function. New evidence (Lin & Zhou, 2020) suggests that considering the *relative action values* leads to better policies, the main argument behind this intuition is that it suffices to identify the optimal actions to solve a task. We extend this principle of relative action to include both state and state-action value functions with a new objective for the critic: the residual variance, *i.e.*, we trade the bias for the variance. In summary, this paper:

- Introduces AVEC, an actor-critic method providing a new training objective for the critic based on the residual variance.
- Provides evidence for the improvement of the value function approximation as well as theoretical consistency of the modified gradient estimator.
- Demonstrates experimentally that AVEC yields a significant performance boost on a set of challenging tasks, including environments with sparse rewards, when coupled with state-of-the-art policy gradient algorithms.
- Provides empirical evidences showing a better fit of the true value function is achieved and substantial stabilization of the gradient.

2 Related Work

Our approach builds on three lines of research, of which we give a quick overview: the use of gradient ascent to update neural network-enabled RL agents, the adoption of a baseline (or control variates) in the gradient estimator, and the introduction of residual variance estimates of Monte Carlo targets.

Policy gradient methods use stochastic gradient ascent to compute a policy gradient estimator. In practice, this is done by computing the gradient over several batches of the sampled trajectory and averaging the outputs (Kakade, 2002). Extensive research have investigated methods to improve the stability of gradient updates. Although it is possible to obtain an unbiased estimate of the policy gradient from empirical trajectories, its variance can be extremely high.

To improve stability, Weaver & Tao (2001) show that subtracting a baseline (Williams, 1992) from the Q-function in the policy gradient can be very beneficial in reducing variance without damaging the bias. Often, variance reduction by introducing a baseline is studied using a critic to approach the state-value function or the Q-function (Silver et al., 2014; Schulman et al., 2016) but such efforts for the actor-critic framework usually result in improved performance without significant effects on variance (Tucker et al., 2018; Ilyas et al., 2020). A combination of these two approaches has led to significant policy gradient algorithms, starting with REINFORCE (Williams, 1988, 1992). Kakade & Langford (2002) later modified policy iteration to create conservative policy iteration, and provided lower bounds for the minimum objective improvement. Peters et al. (2010) then replaced regularization with a trust region constraining policy movement to stabilize training. Currently, state-of-the-art on-policy methods are proximal policy iteration (PPO) (Schulman et al., 2017) and trust region policy optimization (TRPO) (Schulman et al., 2015), both of which require new samples to be collected for each gradient step. Another direction of research that overcomes this limitation is off-policy algorithms, which therefore benefit from all sample transitions; soft actor-critic (SAC) (Haarnoja et al., 2018) is one such approach achieving state-of-the-art performance.

To our knowledge, no previous work applies the variance of residual errors of the value function as an objective function in RL. Although, several works use it as a form of regularization (Jaderberg et al., 2016; Namkoong & Duchi, 2017; Flet-Berliac & Preux, 2019; Kartal et al., 2019).

3 Preliminaries

3.1 Background and Notations

We consider an infinite-horizon Markov Decision Problem (MDP) with continuous states $s \in \mathcal{S}$, continuous actions $a \in \mathcal{A}$, transition distribution $s_{t+1} \sim \mathcal{P}(s_t, a_t)$ and reward function $r_t \sim \mathcal{R}(s_t, a_t)$. Let $\pi_\theta(a|s)$ denote a stochastic policy with parameter θ, we restrict policies to the set of Gaussian distributions. In the following, π and π_θ denote the same object. The agent repeatedly interacts with the environment through sampling actions $a_t \sim \pi(.|s_t)$, receives reward r_t and transitions to a new state s_{t+1}. The objective is to maximize the expected sum of discounted rewards:

$$J(\pi) \triangleq \underset{\tau \sim \pi}{\mathbb{E}}\left[\sum_{t=0}^{\infty} \gamma^{t} r\left(s_{t}, a_{t}\right)\right], \tag{1}$$

where $\gamma \in [0,1)$ is a discount factor (Puterman, 1994), and $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \dots)$ is a trajectory sampled from the environment using policy π. Let us remind the notions of the value of a state s in the MDP framework while following a policy π: $V^{\pi}(s) \triangleq \underset{\tau \sim \pi}{\mathbb{E}}\left[\sum_{t=0}^{\infty} \gamma^{t} r\left(s_{t}, a_{t}\right) | s_0 = s\right]$, the value of a state-action pair: $Q^{\pi}(s,a)$ of performing action a in state s and then following policy π defined as: $Q^{\pi}(s,a) \triangleq \underset{\tau \sim \pi}{\mathbb{E}}\left[\sum_{t=0}^{\infty} \gamma^{t} r\left(s_{t}, a_{t}\right) | s_0 = s, a_0 = a\right]$, and the advantage function which quantifies how an action a is better than the average action in state s: $A^{\pi}(s,a) \triangleq Q^{\pi}(s,a) - V^{\pi}(s)$. In practice, advantage and value functions are unknown; we denote $\hat{A}^{\pi}$, $\hat{V}^{\pi}$, and $\hat{Q}^{\pi}$ their Monte Carlo estimates (as described in Schulman et al. (2016)).

3.2 Deep Policy Gradients

In this section, we consider the case where the value function is learned through a function estimator and then used in an approximation of the gradient. Without loss of generality, we consider the algorithms that approximate the state-value function V, the analysis also holds for algorithms that approximate Q. Let $f_{\phi} : \mathcal{S} \to \mathbb{R}$ be an estimator of $\hat{V}^{\pi}$ with ϕ its parameter and $g_{\phi} = f_{\phi}(s,a) + \mathbb{E}_s[\hat{V}^{\pi}(s) - f_{\phi}(s)]$ its unbiased version. f_{ϕ} is traditionally learned through minimizing the mean squared error (MSE) against $\hat{V}^{\pi}$. At iteration k, the objective of the critic is of the form:

$$\mathcal{L}_{\text{AC}} = \frac{1}{T} \sum_{t=1}^{T} \left[f_{\phi}(s_t) - \hat{V}^{\pi_{\theta_k}}(s_t) \right]^2, \tag{2}$$

where T is the trajectory length. Similarly, one can fit the Monte Carlo estimate $\hat{Q}^{\pi_\theta}$. The policy network optimizes the objective (1) over θ by estimating the gradient (Sutton et al., 2000):

$$\nabla_{\theta} J\left(\pi_{\theta}\right) = \mathbb{E}_{s_t, a_t}\left[\sum_{t=1}^{T} \nabla_{\theta} \log \pi_{\theta}\left(s_{t}, a_{t}\right)\left(\hat{Q}^{\pi_{\theta}}\left(s_{t}, a_{t}\right) - b\left(s_{t}\right)\right)\right], \tag{3}$$

where $b(s_t)$ is a baseline function subtracted to Q to reduce the variance of the gradient without introducing a bias (Greensmith et al., 2004). $b(s_t)$ can be any function of the state s_t. A quasi-optimal choice for reducing the variance of the gradient is $b^*\left(s_t\right) = V^{\pi_\theta}(s_t)$ (Sutton et al., 2000; Mnih et al., 2016). In practice, V^{π} is unknown and g_{ϕ} is used.

4 Method: Actor with Variance Estimated Critic

4.1 Defining an Alternative Critic

Recent work (Ilyas et al., 2020) demonstrate that while the value network succeeds in the supervised learning task of fitting $\hat{V}^{\pi}$, it does not fit V^{π}, and the variance reduction effect is only marginal compared to having access to the true value function. We address this deficiency in the estimation of V^{π} by proposing an alternative value network loss. Following empirical evidence indicating that the problem is the approximation error and not the estimator *per se*, AVEC adopts a loss that can guarantee a better approximation error, and yields better estimators of the value function (as will be shown in Section 5.3):

$$\mathcal{L}_{\text{AVEC}} = \frac{1}{T} \sum_{t=1}^{T} \left[(f_{\phi}(s_t) - \hat{V}^{\pi}(s_t)) - \frac{1}{T} \sum_{t=1}^{T} (f_{\phi}(s_t) - \hat{V}^{\pi}(s_t)) \right]^2. \tag{4}$$

After each training iteration, we center f_{ϕ} on $\hat{V}^{\pi}$ to get the unbiased estimator g_{ϕ}.

4.2 Building Motivation

Tucker et al. (2018) and Ilyas et al. (2020) indicate that the approximation error $\|\hat{V}^\pi - V^\pi\|$ is problematic, suggesting that the variance of the Monte Carlo targets $V^\pi(s_t)$ is high. Optimizing the critic with $\mathcal{L}_{\text{AVEC}}$ can be interpreted as fitting $\hat{V}'^\pi(s_t) = \hat{V}^\pi(s_t) - \frac{1}{T}\sum_{t'=1}^T \hat{V}^\pi(s_{t'})$ using the MSE. We show that the new targets $\hat{V}'^\pi$ are more reliable estimations of $V'^\pi(s_t) = V^\pi(s_t) - \frac{1}{T}\sum_{t'=1}^T V^\pi(s_{t'})$ than $\hat{V}^\pi$ are of V^π. To illustrate this, consider T independent random variables $(X_i)_{i\in\{1,\dots,T\}}$, and note $X'_i = X_i - \frac{1}{T}\sum_{j=1}^T X_j$. We have:

$$\text{Var}\left(X'_i\right) = \text{Var}\left(X_i\right) - \frac{2}{T}\text{Var}\left(X_i\right) + \frac{1}{T^2}\sum_{j=1}^{T}\text{Var}\left(X_j\right).$$

Then $\text{Var}(X'_i) < \text{Var}(X_i)$ as long as $\forall i \ \frac{1}{T}\sum_{j=1}^T \text{Var}(X_j) < 2\text{Var}(X_i)$. Therefore, if state-values are not strongly negatively correlated[1] and not very discordant: $\text{Var}(\hat{V}'_i) < \text{Var}(\hat{V}_i)$. This entails that $\hat{V}'^\pi$ has a more compact span, and is consequently easier to fit. Another intuition using relative values is illustrated in Fig. 1 where grey markers are observations and the blue line is our current estimation. Minimizing the MSE, the line is expected to move towards the orange one in order to reduce errors uniformly. Minimizing the residual variance, it is expected to move near the red one. In fact, $\mathcal{L}_{\text{AVEC}}$ tends to further penalize observations that are far away from the mean, implying that our new loss allows for a better recovery of the "shape" of the target near extrema. In particular, we see in the figure that the maximum and minimum observation values are quickly identified. When the approximators are linear and the $(\hat{V}^\pi(s_t))_{t\in\{1,\dots,T\}}$ are independent, the two losses become equivalent since ordinary least squares provides minimum-variance mean-unbiased estimation.

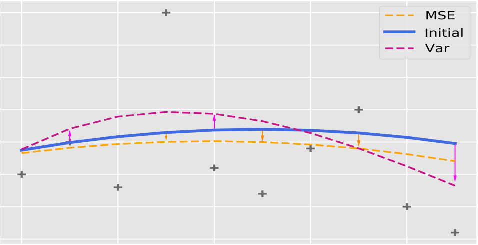


Figure 1: Comparison of simple models derived when $\mathcal{L}_{\text{AVEC}}$ is used instead of the MSE.

When considering a Q-function critic, Eq. (4) translates into replacing $\hat{V}^\pi(s_t)$ by $\hat{Q}^\pi(s_t, a_t)$, and the rationale for optimizing the residual variance of the value function instead of the full MSE becomes more straightforward: the practical use of the Q-function is to disentangle the relative value of the actions for each state (Sutton et al., 2000).

It should be noted that, as in all the works related to ours, we consider deterministic tasks, therefore noise-less tasks. In this context, high estimation errors indicate where (in the state or action-state space) the training of the value function should be improved as opposed to possible outliers.

4.3 Consistency of the New Gradient Estimate

In Appendix A, we demonstrate that our new target interpolation procedure provides consistent gradient directions. As such, when f_ϕ is an estimator of $\hat{Q}^\pi$ and ϕ is updated to minimize $\mathcal{L}_{\text{AVEC}}$, using the parameterization assumption (Sutton et al., 2000) we find:

$$\nabla_\theta J\left(\pi_\theta\right) = \mathbb{E}_{s,a}\left[\nabla_\theta \log(\pi_\theta(s,a)) g_\phi(s,a)\right].$$

4.4 Implementation

We apply this new formulation to three of the most popular deep policy gradient methods to study whether it results in a better estimation of the value function. Given a better estimation, the gradient would be better stabilized and would induce better policy improvements. We now describe how AVEC incorporates its residual variance objective into the critics of PPO (Schulman et al., 2017), TRPO (Schulman et al., 2015) and SAC (Haarnoja et al., 2018). Firstly, line 12 of Algorithm 1 is a

[1] Greensmith et al. (2004) analyze the dependent case. Intuitively, if the MDP is communicating and ergodic, then state values are very similar thus positively correlated.

direct translation of Eq. (4) adapted for PPO or TRPO:

$$\mathcal{L}^{1}_{\text{AVEC}}\left(\phi\right)=\frac{1}{|\mathcal{B}|}\sum_{(s,a,r,s')\in\mathcal{B}}\left[\left(V_{\phi}\left(s\right)-\hat{V}^{\pi}\left(s\right)\right)-\frac{1}{|\mathcal{B}|}\sum_{(s,a,r,s')\in\mathcal{B}}\left(V_{\phi}\left(s\right)-\hat{V}^{\pi}\left(s\right)\right)\right]^{2},$$

where $\hat{V}^{\pi}$ is the empirical value function estimate and $\mathcal{B}$ is a batch of transitions, we unbias $\hat{V}_{\phi}$ after the loss in minimized. Secondly, line 13 of Algorithm 2 (see Appendix B), the loss function of the two Q-functions of SAC when coupled with AVEC translates into:

$$\mathcal{L}^{2}_{\text{AVEC}}\left(\phi_i\right)=\frac{1}{|\mathcal{B}|}\sum_{(s,a,r,s')\in\mathcal{B}}\left[\left(Q_{\phi_i}(s,a)-\hat{Q}^{\pi}(s,a)\right)-\frac{1}{|\mathcal{B}|}\sum_{(s,a,r,s')\in\mathcal{B}}\left(Q_{\phi_i}(s,a)-\hat{Q}^{\pi}(s,a)\right)\right]^{2},$$

where $\hat{Q}^{\pi}$ is the empirical soft Q-function estimate of Q^{π}, $(Q_{\phi_i})_{i=1,2}$ are the two Q-function estimators which we unbias once the training is done and $\mathcal{B}$ is a batch of transitions.

5 Experimental Study

In this section, we conduct experiments in four orthogonal directions. (a) We validate the superiority of AVEC compared to classical actor-critic training. (b) We evaluate whether AVEC succeeds in environments with sparse rewards. (c) We clarify the practical implications of using AVEC by examining the bias in both the empirical and true value function estimations as well as the variance in the empirical gradient. (d) Lastly, we provide an ablation analysis and study the bias-variance trade-off in the critic by considering two continuous control tasks. The code for AVEC is included in the supplementary materials and will be made available online. We point out that a comparison to variance-reduction methods is not considered in this paper: Tucker et al. (2018) demonstrated that their implementations diverge from the unbiased methods presented in the respective papers and unveiled that not only do they fail to reduce the variance of the gradient, but that their unbiased versions do not improve performance either.

Algorithm 1 AVEC coupled with PPO or TRPO. J^{ALGO} denotes the policy loss of either algorithm (described in Schulman et al. (2017, 2015)).

```
Input parameters: $\lambda_\pi \geq 0, \lambda_V \geq 0$
Initialize policy parameter $\theta$ and value function parameter $\phi$
for each update step do
    batch $\mathcal{B} \leftarrow \emptyset$
    for each environment step do
        $a_t \sim \pi_\theta(s_t)$
        $s_{t+1} \sim \mathcal{P}\left(s_t, a_t\right)$
        $\mathcal{B} \leftarrow \mathcal{B} \cup \{(s_t, a_t, r_t, s_{t+1})\}$
    end for
    for each gradient step do
        $\theta \leftarrow \theta - \lambda_\pi \hat{\nabla}_\theta J^{\text{ALGO}}(\pi_\theta)$
        $\phi \leftarrow \phi - \lambda_V \hat{\nabla}_\phi \mathcal{L}^1_{\text{AVEC}}\left(\phi\right)$
    end for
end for
```

5.1 Continuous Control

For ease of comparison with other methods, we evaluate AVEC on the MuJoCo (Todorov et al., 2012) and the PyBullet (Coumans & Bai, 2016) continuous control benchmarks (see Appendix G for details) using OpenAI Gym (Brockman et al., 2016). These tasks are well-established in the RL community and the PyBullet versions of the locomotion tasks are known to be harder than the MuJoCo equivalents. We choose a representative set of tasks for the experimental evaluation; their action and observation space dimensions are reported in Appendix H.

We assess the benefits of AVEC when coupled with the most prominent policy gradient algorithms, currently state-of-the-art methods: TRPO (Schulman et al., 2015) and PPO (Schulman et al., 2017) both on-policy methods, and SAC (Haarnoja et al., 2018), an off-policy maximum entropy deep RL algorithm. For ease of reproducibility, we forked the official *stable-baselines* repository (Hill et al., 2018) and modified the code to incorporate AVEC. We provide the list of hyper-parameters in Appendix C and further experiment details in Appendix D.

Table 1 reports the results while Fig. 2 shows the total average return for SAC and PPO (TRPO results are provided in Appendix E for readability). Although the improvement in performance is less striking for TRPO, AVEC still manages to be more efficient in terms of sampling in all tasks. As for SAC and PPO, it is clear that AVEC brings very significant improvement in the performance

Table 1: Average total reward of the last 100 episodes over 6 runs of 10^6 timesteps. Comparative evaluation of AVEC with SAC and PPO. $\pm$ corresponds to a single standard deviation over trials and $(.\%)$ is the change in performance due to AVEC.

Task	**SAC**	**AVEC-SAC**	**PPO**	**AVEC-PPO**
Ant	3084	$\mathbf{3650 \pm 127}$ $(\mathbf{+18\%})$	972	$\mathbf{1202 \pm 148}$ $(\mathbf{+24\%})$
AntBullet	1193	$\mathbf{2252 \pm 82}$ $(\mathbf{+89\%})$	1174	$\mathbf{2216 \pm 99}$ $(\mathbf{+89\%})$
HalfCheetah	10028	$\mathbf{11018 \pm 102}$ $(\mathbf{+10\%})$	1068	$\mathbf{1403 \pm 37}$ $(\mathbf{+31\%})$
HalfCheetahBullet	1255	$\mathbf{1331 \pm 184}$ $(\mathbf{+6\%})$	1329	$\mathbf{2223 \pm 62}$ $(\mathbf{+67\%})$
Humanoid	4084	$\mathbf{4472 \pm 424}$ $(\mathbf{+10\%})$	391	$\mathbf{415 \pm 4.6}$ $(\mathbf{+6\%})$
Reacher	-6.0	$\mathbf{-5.0 \pm 0.1}$ $(\mathbf{+20\%})$	-7.4	$\mathbf{-5.9 \pm 0.3}$ $(\mathbf{+25\%})$

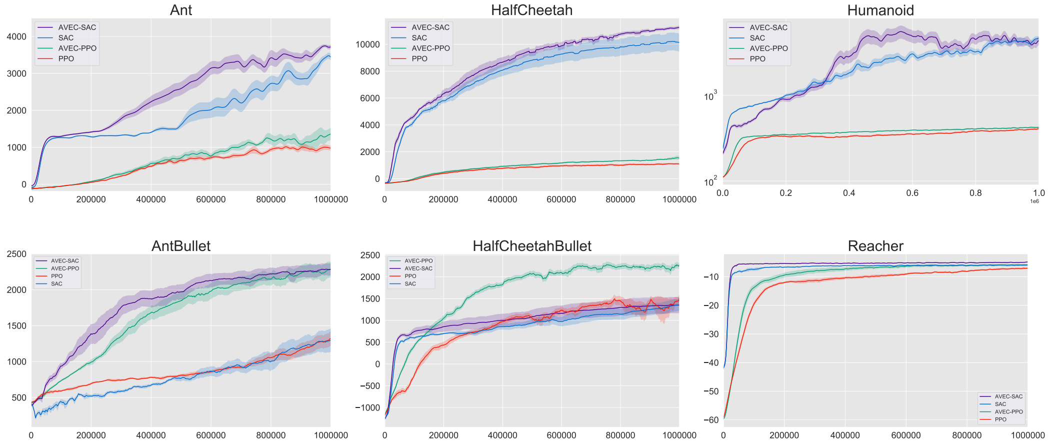
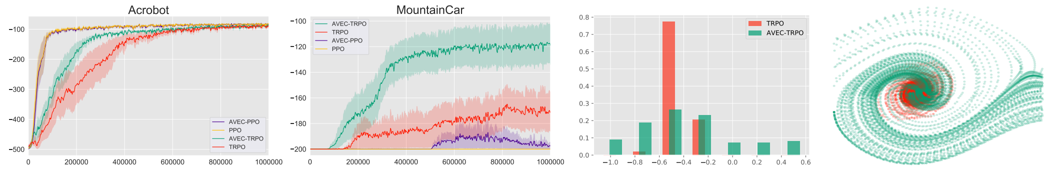


Figure 2: Comparative evaluation (6 seeds) of AVEC with SAC and PPO on PyBullet ("TaskBullet") and MuJoCo ("Task") tasks. X-axis: number of timesteps. Y-axis: average total reward. Lines are average performances and shaded areas represent one standard deviation.

of the policy gradient algorithms. Overall, AVEC outperforms all other policy gradient algorithms that use the traditional actor-critic framework in terms of final performance, and since no additional calculations are needed for its implementation, sample complexity remains unchanged.

5.2 Sparse Reward Signals

By definition, domains with sparse rewards are difficult to solve with uniform exploration as agents receive no feedback on their actions before starting to collect rewards. Under these conditions, AVEC proves to be more performant: as demonstrated in 4.2, AVEC identifies extreme state-values (*e.g.*, non-zero rewards in tasks with sparse reward) faster.



(a) (b) (c) (d)

Figure 3: (a,b): Comparative evaluation (6 seeds) of AVEC with PPO and TRPO on 2 tasks with sparse rewards. X-axis: number of timesteps. Y-axis: average total reward. Lines are average performances and shaded areas represent one standard deviation. (c,d): Respectively state visitation frequency and phase portrait of visited states of AVEC-TRPO (green) and TRPO (red) in MountainCar.

In Fig. 3a and 3b, we report the performance of AVEC in the Acrobot and MountainCar environments: both have extremely sparse rewards. AVEC enhances TRPO and PPO, and when PPO achieves maximal performance, AVEC-PPO improves sample complexity. Fig. 3c and 3d illustrate how the agent improves its exploration strategy in MountainCar: while the PPO agent remains stuck at the bottom of the hill (red), the graph suggest that AVEC-PPO learns the difficult locomotion principles in the absence of rewards and visits a much larger part of the state space (green).

5.3 Variance Estimated Critic Impact Assessment

In order to further validate AVEC, we analyze the performance of the value network: we examine (a) the estimation error (distance to the empirical target), (b) the approximation error (distance to the true target) and (c) the empirical variance of the gradient. (a,b) should be put into perspective with the conclusions of Ilyas et al. (2020) where it is found that the critic fits the empirical value function but not the true one. (c) should be placed in light of Tucker et al. (2018) highlighting a failure of recently proposed state-action-dependent baselines to reduce the variance.

Learning the Empirical Target. In Fig. 4, we report the quality of fit (MSE) of $\hat{V}^\pi$ by PPO and AVEC-PPO in the AntBullet and HalfCheetahBullet tasks. We observe that PPO fits the empirical target much better than when coupled with AVEC, which was expected since vanilla PPO optimizes the MSE directly. This result and the remarkable improvement in the performance of AVEC-PPO (Fig. 2) strongly suggest that AVEC is a better estimator of the true value function. If true, this would indicate that it is possible to simultaneously improve the performance of the agents and the stability of the method. We validate this claim in the next section.

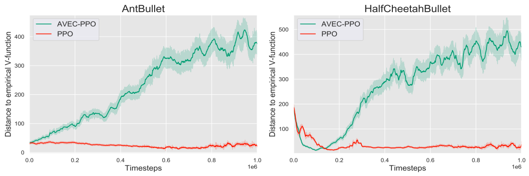


Figure 4: Distance to the empirical value function. Lines are average performances and shaded areas represent one standard deviation.

Learning the True Target. A fundamental premise of policy gradient methods is that optimizing an empirical estimation of the objective leads to a better policy (*cf.* Eq. (3)). For methods using a baseline, this implies that the baseline estimator (typically computed with on the order of 10^3 samples) should be unbiased in order to correctly estimate the gradient. For methods such as SAC, the value estimator directly appears in the objective. Hence, in this section, we investigate the quality of fit of the true target. In order to approximate the true value function, we fit the returns sampled from the current policy using a large number of transitions (3.10^5).

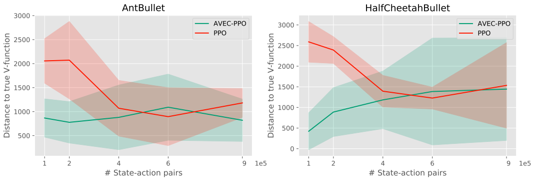


Figure 5: Distance to the true value function. X-axis: we run the algorithm and for every $t \in \{1, 2, 4, 6, 9\}.10^5$ we stop training, use the current policy to collect 3.10^5 transitions, and use these transitions to estimate the true value function. Lines are average performances and shaded areas represent one standard deviation.

Fig. 5 shows the L_2 distance between the value function estimator and the true value function. First, we note that AVEC provides a better estimator of V^π; second, comparing Fig. 5 with the results in Fig. 4, we see that the distance to the true target is close to the estimation error for AVEC-PPO, while for PPO, it is at least two orders of magnitude higher at all times. A similar analysis for the Q-function estimator for SAC and AVEC-SAC in AntBullet and HalfCheetahBullet is provided in Appendix F.1, with similar results. We conclude that the improved approximation error of AVEC not only compensates for the worse estimation error of AVEC but greatly improves the value function approximation. We expect that this would lead to a more stabilized gradient.

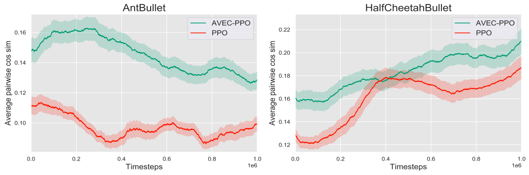
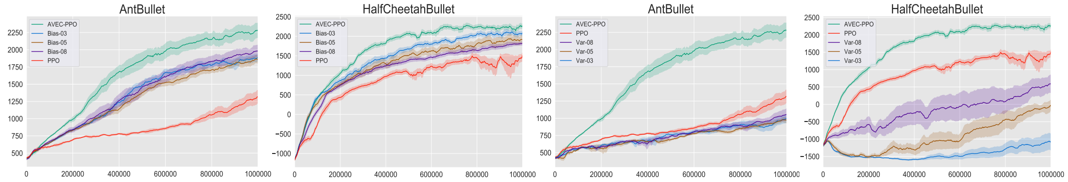


Figure 6: Average cosine similarity between gradient measurements. Lines are average performances and shaded areas represent one standard deviation.

Empirical Variance Reduction. In Fig. 6, we study the empirical variance of the gradient by measuring the average pairwise cosine similarity between gradient measurements (10 batches at each iteration). We observe a consistent reduction in variance when using AVEC, confirming previous results. Further analysis with additional experiments and algorithms are included in Appendix F.2. This variance reduction effect confirmed over multiple environments proves that AVEC is the first method since the introduction of the value function baseline which further reduces the variance of the gradient and improves the performance.

5.4 Ablation Study

In this section, we examine how changing the relative importance of the bias and the residual variance in the loss of the value network affects learning. For this study, we choose difficult tasks of PyBullet and use the method PPO which is significantly more sample efficient than TRPO, and less computationally demanding compared to SAC. For an estimator $\hat{y}_n$ of $(y_i)_{i\in\{1,\dots,n\}}$, we write $\text{Bias} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)$ and $\text{Var} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i - \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i))^2$. Consequently: $\text{MSE} = \text{Var} + \text{Bias}^2$. We denote $\mathcal{L}_\alpha = \text{Var} + \alpha\text{Bias}^2$, with $\alpha \in \mathbb{R}$. In Fig. 7, *Bias*-α means that we use PPO with $\mathcal{L}_\alpha$ and *Var*-α means that we use PPO with $\mathcal{L}_{\frac{1}{\alpha}}$.



(a) (b) (c) (d)

Figure 7: Sensitivity (6 seeds) of AVEC-PPO with respect to (a,b): the bias; (c,d): the variance. X-axis: number of timesteps. Y-axis: average total reward. Lines are average performances and shaded areas represent one standard deviation.

We observe that while no consistent pattern governing the position of the learning curves is identified, AVEC seems to outperform all other weightings. A more extensive study with more α values might provide more intuition, but we believe that the stability of an algorithm is crucial for reliable performance. As such, the tuning of hyper-parameters to achieve good results should remain mild.

6 Discussion

In this work, we introduce a new training objective for the critic in actor-critic algorithms to better approximate the true value function. In addition to being well-motivated by recent studies on the behaviour of deep policy gradient algorithms, we demonstrate that this modification is both theoretically sound and intuitively supported by the need to improve the approximation error of the critic. The application of AVEC (Actor with Variance Estimated Critic) to the most dominant policy gradient methods produces considerable gains in performance over the standard actor-critis training, without any additional hyper-parameter tuning.

First, for SAC-like methods where the critic learns the state-action value function, our results suggest that the relative values of the Q-function are better fitted in extreme regions. Second, for PPO-like methods where the critic learns the state values, we show that the variance of the gradient is reduced and empirically demonstrate that this is due to a better approximation of the state values. In sparse reward environments, the theoretical intuition behind a variance estimated critic is more explicit and is also supported by empirical evidence. In addition to corroborating the results of Ilyas et al. (2020) proving that the value estimator fails to fit the value function, we propose a method that succeeds in improving both the sample complexity and the stability of actor critic algorithms without the need for further assumptions (such as horizon awareness as in Tucker et al. (2018)) to remedy the deficiency of existing variance-reduction methods.

In this paper, we have demonstrated the benefits of a more thorough analysis of the critic objective in policy gradient methods. Despite our strongly favourable results, we do not claim that the residual variance is the best possible loss for the critic, and we note that the design of comparably superior estimators for baselines in deep policy gradient methods merits further study.

Broader Impact

Although we do not consider our work as a direct and imminent potential threat to the balance of our societies and human existence, we will try to identify some reflections that could be relevant to our work. We will also be careful to avoid any speculation.

Understanding the methods we develop is a central point. The principles of interpretable machine learning is an area of direct importance (Zeiler & Fergus, 2014; Ribeiro et al., 2016). This will not only help us to anticipate problems and potential threats related to these methods if implemented in stand-alone machines, but it will certainly also help us to build better machines. To move towards more reliable and efficient methods, we need to dissect and understand each component of the algorithm. Our work, which builds on the work of Tucker et al. (2018); Ilyas et al. (2020) demonstrating a significant gap between the theory of current algorithms and the actual mechanisms that determine their performance, contributes to this path by introducing a new procedure to train the critic in these algorithms that is theoretically sound, improves performance, and is closer to our expectations when examining the learning of the critic.

Moreover, our work has the ambition to further exploit the potential of policy gradient algorithms by restricting the setting. Our improvement consists in defining a better objective function. The objective function is a paradigm at the center of RL algorithms and more broadly at the center of ML. A good objective function makes a good model. “Good” can have several meanings, so after defining what our broader objectives should be, the objective function should be carefully adapted to target the RL agent to these objectives. In our work, we have shown that if the objective function does not fit what it is supposed to fit, a better design is possible, and can at the same time better meet our expectations (better fit the “true” value function) while improving performance.

References

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym, 2016. arXiv:1606.01540.

Coumans, E. and Bai, Y. Pybullet, a python module for physics simulation for games, robotics and machine learning, 2016.

Flet-Berliac, Y. and Preux, P. Merl: Multi-head reinforcement learning. In *Deep Reinforcement Learning Workshop, NeurIPS*, 2019.

Greensmith, E., Bartlett, P. L., and Baxter, J. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov):1471–1530, 2004.

Gu, S., Lillicrap, T., Sutskever, I., and Levine, S. Continuous deep q-learning with model-based acceleration. In *Proc. International Conference on Machine Learning*, pp. 2829–2838, 2016.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proc. International Conference on Machine Learning*, pp. 1856–1865, 2018.

Hill, A., Raffin, A., Ernestus, M., Gleave, A., Kanervisto, A., Traore, R., Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., and Wu, Y. Stable baselines, 2018.

Ilyas, A., Engstrom, L., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., and Madry, A. A closer look at deep policy gradients. In *International Conference on Learning Representations*, 2020.

Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., and Kavukcuoglu, K. Reinforcement learning with unsupervised auxiliary tasks. *arXiv:1611.05397*, 2016.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *Proc. International Conference on Machine Learning*, pp. 267–274, 2002.

Kakade, S. M. A natural policy gradient. In *Proc. Advances in Neural Information Processing Systems*, pp. 1531–1538, 2002.

Kartal, B., Hernandez-Leal, P., , and Taylor, M. E. Terminal prediction as an auxiliary task for deep reinforcement learning. In *Proc. AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, pp. 38–44, 2019.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. In *Proc. International Conference on Learning Representations*, 2016.

Lin, K. and Zhou, J. Ranking policy gradient. In *International Conference on Learning Representations*, 2020.

Liu, H., Feng, Y., Mao, Y., Zhou, D., Peng, J., and Liu, Q. Action-dependent control variates for policy optimization via stein identity. In *Proc. International Conference on Learning Representations*, 2018.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *Proc. International Conference on Machine Learning*, pp. 1928–1937, 2016.

Namkoong, H. and Duchi, J. C. Variance-based regularization with convex objectives. In *Proc. Advances in Neural Information Processing Systems*, pp. 2971–2980, 2017.

Peters, J., Mulling, K., and Altun, Y. Relative entropy policy search. In *Proc. AAAI Conference on Artificial Intelligence*, 2010.

Puterman, M. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.

Ribeiro, M. T., Singh, S., and Guestrin, C. " why should i trust you?" explaining the predictions of any classifier. In *Proc. of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *Proc. International Conference on Machine Learning*, pp. 1928–1937, 2015.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. In *Proc. International Conference on Learning Representations*, 2016.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms, 2017. arXiv:1707.06347.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. Deterministic policy gradient algorithms. In *Proc. International Conference on Machine Learning*, 2014.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Proc. Advances in Neural Information Processing Systems*, 2000.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012.

Tucker, G., Bhupatiraju, S., Gu, S., Turner, R., Ghahramani, Z., and Levine, S. The mirage of action-dependent baselines in reinforcement learning. In *Proc. International Conference on Machine Learning*, pp. 5015–5024, 2018.

Weaver, L. and Tao, N. The optimal reward baseline for gradient·based reinforcement learning. In *Proc. Advances in Neural Information Processing Systems*, 2001.

Williams, R. Toward a theory of reinforcement-learning connectionist systems. *Technical Report NU-CCS-88-3*, 1988.

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

Wu, C., Rajeswaran, A., Duan, Y., Kumar, V., Bayen, A. M., Kakade, S., Mordatch, I., and Abbeel, P. Variance reduction for policy gradient with action-dependent factorized baselines. In *Proc. International Conference on Learning Representations*, 2018.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *13th European Conference on Computer Vision, ECCV 2014*, pp. 818–833. Springer Verlag, 2014.

A Consistency of the AVEC Estimator

In this section, we consider the case in which the state-action value function of a policy π_θ is approximated. We prove that given some assumptions on this estimator function, we can use it to yield a valid gradient direction, *i.e.*, we are able to prove policy improvement when following this direction.

In this setting, the critic minimizes the following loss:

$$\mathbb{E}_{s,a}\left[(\hat{Q}^\pi_\theta(s,a) - f_\phi(s,a) - \mathbb{E}_{s,a}[\hat{Q}^\pi_\theta(s,a) - f_\phi(s,a)])^2\right].$$

When a local optimum is reached, the gradient of the latter expression is zero:

$$\nabla_\phi \mathcal{L}_{\text{AVEC}} = \mathbb{E}_{s,a}\left[(\hat{Q}^\pi_\theta(s,a) - f_\phi(s,a) - \mathbb{E}_{s,a}[\hat{Q}^\pi_\theta(s,a) - f_\phi(s,a)])(\frac{\partial f_\phi(s,a)}{\partial \phi} - \mathbb{E}_{s,a}[\frac{\partial f_\phi(s,a)}{\partial \phi}])\right] = 0.$$

In the expression above, the expected value of the partial derivative disappears because the term in the first bracket is centered:

$$\begin{aligned}
&\mathbb{E}_{s,a}\left[(\hat{Q}^\pi_\theta(s,a) - f_\phi(s,a) - \mathbb{E}_{s,a}[\hat{Q}^\pi_\theta(s,a) - f_\phi(s,a)])\mathbb{E}_{s,a}[\frac{\partial f_\phi(s,a)}{\partial \phi}]\right] \\
&= \mathbb{E}_{s,a}\left[\frac{\partial f_\phi(s,a)}{\partial \phi}\right] \underbrace{\mathbb{E}_{s,a}[\hat{Q}^\pi_\theta(s,a) - f_\phi(s,a) - \mathbb{E}_{s,a}[\hat{Q}^\pi_\theta - f_\phi]]}_{= 0} \\
&= 0.
\end{aligned}$$

Simplifying the gradient at the local optimum becomes:

$$\mathbb{E}_{s,a}\left[(\hat{Q}^\pi_\theta(s,a) - f_\phi(s,a) - \mathbb{E}_{s,a}[\hat{Q}^\pi_\theta(s,a) - f_\phi(s,a)])(\frac{\partial f_\phi(s,a)}{\partial \phi})\right] = 0. \tag{5}$$

Then, if we denote $g_\phi = f_\phi(s,a) + \mathbb{E}_{s,a}[\hat{Q}^\pi(s,a) - f_\phi(s,a)]$, and use the policy parameterization assumption:

$$\frac{\partial f_\phi(s,a)}{\partial \phi} = \frac{\partial \pi_\theta(s,a)}{\partial \theta}\frac{1}{\pi_\theta(s,a)}, \tag{6}$$

we obtain:

$$\boxed{\nabla_\theta J = \mathbb{E}_{s,a}\left[\nabla_\theta \log(\pi_\theta(s,a)) g_\phi(s,a)\right].} \tag{7}$$

Proof. By combining the parameterization assumption in Eq. (6) with Eq. (5), we have:

$$\mathbb{E}_{s,a}\left[(\hat{Q}^\pi_\theta(s,a) - g_\phi(s,a))\frac{\partial \pi_\theta(s,a)}{\partial \theta}\frac{1}{\pi_\theta(s,a)}\right] = 0. \tag{8}$$

Since the expression above is null, we have the following:

$$\begin{aligned}
\nabla_\theta J &= \mathbb{E}_{s,a}[\nabla_\theta \log(\pi_\theta(s,a))\hat{Q}^{\pi_\theta}(s,a)] \\
&= \mathbb{E}_{s,a}[\nabla_\theta \log(\pi_\theta(s,a))\hat{Q}^{\pi_\theta}(s,a)] - \mathbb{E}_{s,a}[(\hat{Q}^\pi_\theta(s,a) - g_\phi(s,a))\frac{\partial \pi_\theta(s,a)}{\partial \theta}\frac{1}{\pi_\theta(s,a)}] \\
&= \mathbb{E}_{s,a}[\nabla_\theta \log(\pi_\theta(s,a)) g_\phi(s,a)].
\end{aligned}$$

□

Remark. While the proof seems more or less generic, the assumption in Eq. (6) is extremely constraining to the possible approximators. Sutton et al. (2000) quotes *J. Tsitsiklis* who believes that a linear g_ϕ in the features of the policy may be the only feasible solution for this condition. Concretely, such an assumption cannot hold since neural networks are the standard approximators used in practice. Moreover, empirical analysis (Ilyas et al., 2020) indicates that commonly used algorithms fail to fit the true value function. However, this does not rule out the usefulness of the approach but rather begs for more questioning of the true effect of such biased baselines.

B Implementation of AVEC coupled with SAC

In Algorithm 2, J_V is the squared residual error objective to train the soft value function. See Haarnoja et al. (2018) for further details and notations about SAC, not directly relevant here.

Algorithm 2 AVEC coupled with SAC.

```
**Input parameters:** $\beta \in [0,1], \lambda_V \geq 0, \lambda_Q \geq 0, \lambda_\pi \geq 0$
**Initialize** policy parameter $\theta$, value function parameter $\psi$ and $\bar{\psi}$ and Q-functions parameters $\phi_1$ and $\phi_2$
$\mathcal{D} \leftarrow \emptyset$
**for** each iteration **do**
    **for** each step **do**
        $a_t \sim \pi_\theta(a_t|s_t)$
        $s_{t+1} \sim \mathcal{P}(s_t, a_t)$
        $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, r_t, s_{t+1})\}$
    **end for**
    **for** each gradient step **do**
        sample batch $\mathcal{B}$ from $\mathcal{D}$
        $\psi \leftarrow \psi - \lambda_V \hat{\nabla}_\psi J_V(\psi)$
        $\phi_i \leftarrow \phi_i - \lambda_Q \hat{\nabla}_{\phi_i} \mathcal{L}^2_{\text{AVEC}}(\phi_i)$ for $i \in \{1,2\}$
        $\theta \leftarrow \theta - \lambda_\pi \hat{\nabla}_\theta J(\pi_\theta)$
        $\bar{\psi} \leftarrow \beta\psi + (1-\beta)\bar{\psi}$
    **end for**
**end for**
```

C Implementation Details

C.1 Source Code

The source code is available in the supplementary materials. A “README.md” file is included with basic instructions on how to execute the code.

C.2 Training Infrastructure

We ran all our experiments using 10 nodes, each one with dual Intel Xeon Gold 6126 CPUs and Nvidia V100 or P100 GPUs.

C.3 Hyper-parameters

In Table 2, 3 and 4, we report the list of hyper-parameters common to all continuous control experiments.

Table 2: Hyper-parameters used both in SAC and AVEC-SAC.

Parameter	Value
Adam stepsize	$3 \cdot 10^{-4}$
Discount (γ)	0.99
Replay buffer size	10^6
Learning starts	10000
Batch size	256
Nb. hidden layers	2
Nb. hidden units per layer	256
Nonlinearity	ReLU
Target smoothing coefficient (τ)	0.01
Target update interval	1
Gradient steps	1

Table 3: Hyper-parameters used both in PPO and AVEC-PPO.

Parameter	Value
Horizon (T)	2048
Adam stepsize	$2.5 \cdot 10^{-4}$
Nb. epochs	10
Nb. minibatches	32
Nb. hidden layers	2
Nb. hidden units per layer	64
Nonlinearity	tanh
Discount (γ)	0.99
GAE parameter (λ)	0.95
Clipping parameter (ϵ)	0.2

Table 4: Hyper-parameters used both in TRPO and AVEC-TRPO.

Parameter	Value
Horizon (T)	2048
Adam stepsize	$1 \cdot 10^{-4}$
Nb. hidden layers	2
Nb. hidden units per layer	64
Nonlinearity	tanh
Discount (γ)	0.99
GAE parameter (λ)	0.95
Stepsize KL	0.01
Nb. iterations for the conjugate gradient	15

D Experiment Details

In all experiments we choose to use the same hyper-parameter values (see Appendix C.3) for all tasks as the best-performing ones reported in the literature or in their respective open source implementation documentation. We thus ensure the best performance for the conventional actor-critic framework. In other words, since we are interested in evaluating the impact of this new critic, everything else is kept as is. This experimental protocol may not be the best for AVEC.

E Comparative Evaluation of AVEC with TRPO

In order to evaluate the performance gains in using AVEC instead of the usual actor-critic framework, we produce some additional experiments with the TRPO (Schulman et al., 2015) algorithm. Fig. 8 shows the learning curves while Table 5 reports the results.

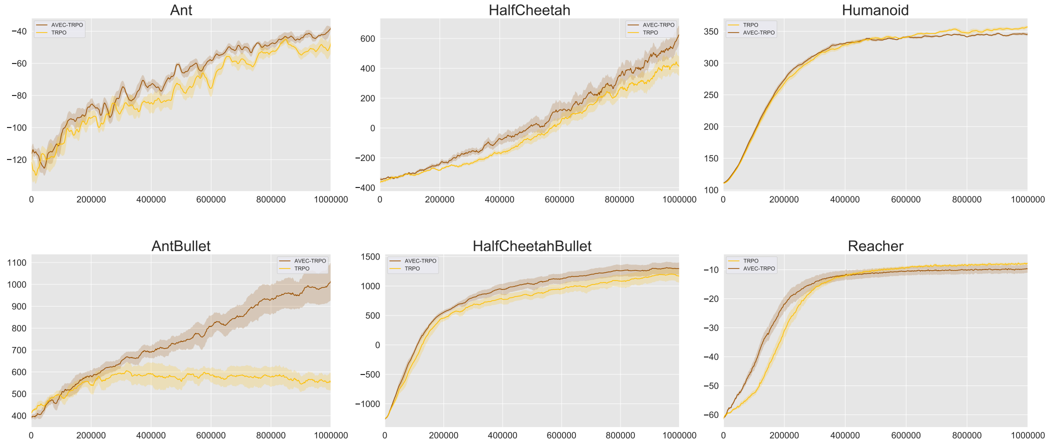


Figure 8: Comparative evaluation of AVEC with TRPO. We run with 6 different seeds: lines are average performances and shaded areas represent one standard deviation.

Table 5: Average total reward of the last 100 episodes over 6 runs of 10^6 timesteps. Comparative evaluation of AVEC with TRPO. ± corresponds to a single standard deviation over trials and $(.\%)$ is the change in performance due to AVEC.

Task	**TRPO**	**AVEC-TRPO**
Ant	-50.5	$\mathbf{-43.5 \pm 2.2}\,(\mathbf{+16\%})$
AntBullet	564	$\mathbf{970 \pm 70}\,(\mathbf{+72\%})$
HCheetah	346	$\mathbf{466 \pm 56}\,(\mathbf{+35\%})$
HCBullet	1154	$\mathbf{1281 \pm 94}\,(\mathbf{+11\%})$
Humanoid	$\mathbf{352}$	$344 \pm 1.2\,(-3\%)$
Reacher	$\mathbf{-8.5}$	$-9.9 \pm 1.3\,(-16\%)$

F Additional Experiments

F.1 Learning the True Target: SAC

In Fig. 9, we compare the error between the Q-function estimator and the true Q-function for SAC and AVEC-SAC in AntBullet and HalfCheetahBullet. We note a modest but consistent reduction in this error when using AVEC coupled with SAC, echoing the significant performance gains in Fig. 2.

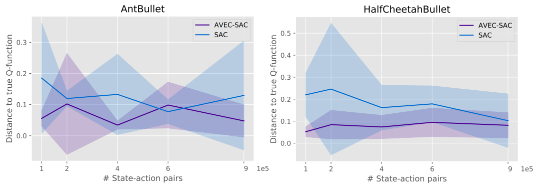


Figure 9: Distance to the true Q-function (SAC). X-axis: we run the algorithm and for every $t \in \{1, 2, 4, 6, 9\}.10^5$ we stop training, use the current policy to interact with the environment for 3.10^5 transitions, and use these transitions to estimate the true value function. Lines are average performances and shaded areas represent one standard deviation.

F.2 Variance Reduction

In Fig. 10, we study the empirical variance of the gradient in measuring the average pairwise cosine similarity (10 gradient measurements) in two additional tasks: HopperBullet and Walker2DBullet. We also vary the trajectory size used in the estimation of the gradient.

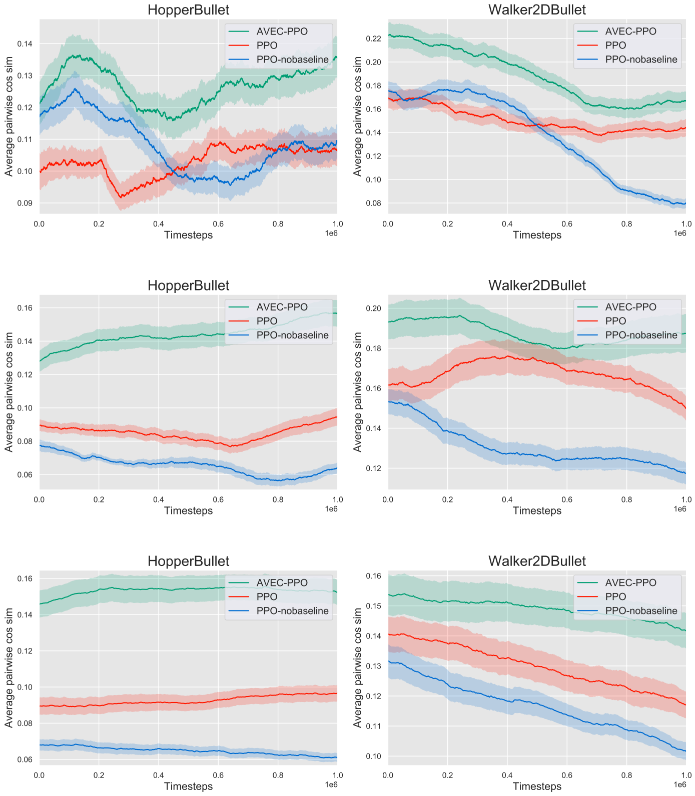


Figure 10: Average cosine similarity between gradient measurements. AVEC empirically reduces the variance compared to PPO or PPO without a baseline (PPO-nobaseline). Trajectory size used in estimation of the gradient variance: 3000 (upper row), 6000 (middle row), 9000 (lower row). Lines are average performances and shaded areas represent one standard deviation.

G Environments Details

Table 6: Environments details.

Environment	Description
Ant-v2	Make a four-legged creature walk forward as fast as possible.
AntBulletEnv-v0	Idem. Ant is heavier, encouraging it to typically have two or more legs on the ground (source: PyBullet Guide - url).
HalfCheetah-v2	Make a 2D cheetah robot run.
HalfCheetahBulletEnv-v0	Idem.
Humanoid-v2	Make a three-dimensional bipedal robot walk forward as fast as possible, without falling over.
Reacher-v2	Make a 2D robot reach to a randomly located target.
Acrobot-v1	Swing the end of a two-joint acrobot up to a given height.
MountainCar-v0	Get an under powered car to the top of a hill.

H Dimensions of Studied Tasks

Table 7: Actions and observations dimensions.

Task	$\mathcal{S}$	$\mathcal{A}$
Ant	$\mathbb{R}^{111}$	$\mathbb{R}^{8}$
AntBullet	$\mathbb{R}^{28}$	$\mathbb{R}^{8}$
HalfCheetah	$\mathbb{R}^{17}$	$\mathbb{R}^{6}$
HalfCheetahBullet	$\mathbb{R}^{26}$	$\mathbb{R}^{6}$
Humanoid	$\mathbb{R}^{376}$	$\mathbb{R}^{17}$
Reacher	$\mathbb{R}^{11}$	$\mathbb{R}^{2}$
Acrobot	$\mathbb{R}^{6}$	3
MountainCar	$\mathbb{R}^{2}$	3