Learning Value Functions using Residual Variance in Deep Policy Gradients

Yannis Flet-Berliac*¹,², based on work with:
Reda Ouhamma*¹,², Odalric-Ambrym Maillard¹, Philippe Preux¹,²

¹SequeL team – Inria Lille – Nord Europe, France
²Univ. Lille, CRIStAL, CNRS
*Equal contribution
Agenda

- Policy Gradients (PG) and Actor-Critic (AC) methods
  - Contextualization
  - The use of value functions (Critics) in deep PG algorithms
- Problem: empirical failure of popular AC methods
  - Critics do not actually fit $V^\pi$
  - State-action-dependent baselines fail to reduce gradient variance
- AVEC: learning the Critics using Residual Variance
  - An alternative Critic (+ building motivation)
  - Consistent gradient directions
  - Empirical results: continuous control + sparse-reward tasks
- AVEC: does it really work?
  - Estimation error (learning the empirical target $\hat{V}^\pi$)
  - Approximation error (learning the true target $V^\pi$)
  - Empirical variance
- Conclusion and perspectives
Reinforcement Learning

Environment (Markov Decision Process):

- State $s \in S$, action $a \in A$
- Reward function: $r(s, a)$, transition probabilities: $P(s' | s, a)$

Agent:

- Stochastic policy $\pi_\theta(a | s)$ with parameter $\theta$

An agent in state $s_t$ interacts with an environment by sampling action $a_t \sim \pi_\theta(\cdot | s_t)$, receives reward $r_t$ and transitions to a new state $s_{t+1}$. 
Reinforcement Learning

Environment (Markov Decision Process):
- State $s \in S$, action $a \in A$
- Reward function: $r(s, a)$, transition probabilities: $P(s' | s, a)$

Agent:
- Stochastic policy $\pi_\theta(a | s)$ with parameter $\theta$

An agent in state $s_t$ interacts with an environment by sampling action $a_t \sim \pi_\theta(\cdot | s_t)$, receives reward $r_t$ and transitions to a new state $s_{t+1}$.

Goal: Find $\pi$ that maximizes

$$J(\pi_\theta) \triangleq \mathbb{E}_{\tau \sim \pi_\theta} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]$$

with $\gamma \in [0, 1)$, $s_{t+1} \sim P(\cdot | s_t, a_t)$, $a_t \sim \pi_\theta(\cdot | s_t)$ and trajectory $\tau$. 

Revisiting the Critics in Deep Policy Gradients
Policy Gradients

Policy gradient algorithms try to solve the optimization problem

$$\max_\theta J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]$$

by taking stochastic gradient ascent on the policy parameters $\theta$, using the policy gradient

$$\nabla_\theta J = \mathbb{E}_{\tau \sim \pi_\theta} \left[ \sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t|s_t) Q^{\pi_\theta}(s_t, a_t) \right]$$

with $Q^{\pi_\theta}(s, a) = \mathbb{E}_{\tau \sim \pi_\theta} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a \right]$.

**Intuition:** make the good actions more probable.
Policy Gradients

It is possible to obtain an unbiased estimate of the policy gradient from empirical trajectories ...

... But the corresponding variance can be extremely high.
Policy Gradients

It is possible to obtain an unbiased estimate of the policy gradient from empirical trajectories ...

... But the corresponding variance can be extremely high.

Subtracting a baseline from the value function in the policy gradient can be very beneficial in reducing variance without damaging the bias [Williams, 1992, Weaver and Tao, 2001].
Policy Gradients

It is possible to obtain an unbiased estimate of the policy gradient from empirical trajectories ...

... But the corresponding **variance can be extremely high**.

Subtracting a baseline from the value function in the policy gradient can be very beneficial in reducing variance without damaging the bias [Williams, 1992, Weaver and Tao, 2001].

In practice, if we denote $\hat{X}$ the empirical estimate of $X$, the policy gradient becomes

$$\nabla_\theta J = \mathbb{E}_{\tau \sim \pi_\theta} \left[ \sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t|s_t) \hat{A}^\pi_\theta(s_t, a_t) \right],$$

with $\hat{A}^\pi_\theta(s, a) = \hat{Q}^\pi_\theta(s, a) - \hat{V}^\pi_\theta(s)$ the advantage estimate which quantifies how an action $a$ is better than the average action in state $s$. 
Critics in Deep Policy Gradients

\( \hat{V}^{\pi_\theta} \) is learned using a function estimator.

Let \( f_\phi : S \rightarrow \mathbb{R} \) (\( \phi \) its parameter) be an estimator of the empirical return \( \hat{V}^{\pi_\theta} \). \( f_\phi \) is traditionally learned through minimizing the MSE against \( \hat{V}^{\pi_\theta} \). At update \( k \), the critic minimizes:

\[
L_{AC} = \mathbb{E}_s \left[ (f_\phi(s) - \hat{V}^{\pi_\theta_k}(s))^2 \right],
\]

where the states \( s \) are collected under policy \( \pi_{\theta_k} \).

Setting applicable to e.g. PPO [Schulman et al., 2017] and TRPO [Schulman et al., 2015]. Alternatively use \( f_\phi : S \times A \rightarrow \mathbb{R} \) instead to fit \( \hat{Q}^{\pi_\theta} \) to use in SAC [Haarnoja et al., 2018].
The sad truth about policy gradient variance

In practice, these modifications on the AC framework result in improved performance without a significant variance reduction [Ilyas et al., 2019]. Same conclusion with state-action-dependent baselines [Tucker et al., 2018].
The sad truth about policy gradient variance

In practice, these modifications on the AC framework result in improved performance without a significant variance reduction [Ilyas et al., 2019]. Same conclusion with state-action-dependent baselines [Tucker et al., 2018].

**Problem:** discrepancy between what motivates AC algorithms and the resulting implementation to obtain maximum gains.

**Figure:** Empirical variance of the gradient [Ilyas et al., 2019].

NB: the avg. pairwise cos sim is inversely proportional to the gradient variance (the higher the better).
The sad truth about learning the value function

The value network (critic) succeeds in the supervised learning task of fitting $\hat{V}^\pi$ but does not fit $V^\pi$.

→ The problem is the approximation error and not the estimator.

Figure: Quality of value prediction [Ilyas et al., 2019]
The sad truth about learning the value function

- Variance **is not reduced**.
- Approximation error: the value network succeeds in fitting $\hat{V}^\pi$ but **does not fit** $V^\pi$.
- Unbiased equivalents of “novel baseline-based algorithms” **do not improve performance nor reduce variance** (see [Tucker et al., 2018]).
A new approach to learning value functions

In AVEC, at update $k$ the critic minimizes:

$$
\mathcal{L}_{AC}(\phi) = \mathbb{E}_s \left[ (f_{\phi}(s) - \hat{V}_{\pi_{\theta_k}}(s))^2 \right]
$$

$$
\mathcal{L}_{AVEC}(\phi) = \mathbb{E}_s \left[ \left( (f_{\phi}(s) - \hat{V}_{\pi_{\theta_k}}(s)) - \mathbb{E}_s \left[ f_{\phi}(s) - \hat{V}_{\pi_{\theta_k}}(s) \right] \right)^2 \right]
$$

where the states $s$ are collected under policy $\pi_{\theta_k}$.

→ **AVEC** minimizes the residual variance.

Using $g_{\phi}(s) = f_{\phi}(s) + \mathbb{E}_s[\hat{V}_{\pi_{\theta_k}}(s) - f_{\phi}(s)]$ provides consistent gradient directions:

$$
\nabla_{\theta} J(\pi_{\theta}) = \mathbb{E}_{\tau \sim \pi_{\theta}} \left[ \sum_{t=0}^{\infty} \nabla_{\theta} \log \pi_{\theta}(a_t|s_t) g_{\phi}(s_t) \right].
$$

**Proof.** See [Flet-Berliac et al., 2020].
Intuition

The most straightforward intuition comes in the Q-function estimation case: simply replace $\hat{V}^\pi(s)$ by $\hat{Q}^\pi(s, a)$ and $f_\phi(s)$ by $f_\phi(s, a)$ in $L_{\text{AVEC}}$.

Idea: the practical use of the Q-function is to disentangle the relative values of actions for each state [Sutton et al., 2000]. $L_{\text{AVEC}}$ focuses on the relative state-action values.

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

→ In minimizing the residual variance, AVEC allows a better recovery of the “shape” of the target near extrema.
Recall the approximation error $\| \hat{V}^\pi - V^\pi \|$ is problematic [Tucker et al., 2018, Ilyas et al., 2019].
$\rightarrow$ Suggests that the variance of the empirical targets $V^\pi$ is high.

**Idea:** optimizing the critic with $\mathcal{L}_{\text{AVEC}}$ can be interpreted as fitting

$$\hat{V}'^\pi(s) = \hat{V}^\pi(s) - \mathbb{E}_{s'}[\hat{V}^\pi(s')]$$

using the MSE.

In the independent case, $\hat{V}'^\pi$ is a better approximation of $V'^\pi(s) = V^\pi(s) - \mathbb{E}_{s'}[V^\pi(s')]$ than $\hat{V}^\pi$ is of $V^\pi$ (see [Flet-Berliac et al., 2020]).

$\rightarrow$ $\hat{V}'^\pi$ has a more compact span, and is consequently **easier to fit**.
## Implementation

**Algorithm 1** AVEC coupled with PPO.

1. **Input parameters:** \( \lambda_\pi \geq 0, \lambda_V \geq 0 \)
2. **Initialize** policy parameter \( \theta \) and value function parameter \( \phi \)
3. **for** each update step **do**
4. \( \text{batch } B \leftarrow \emptyset \)
5. **for** each environment step **do**
6. \( a_t \sim \pi_\theta(\cdot|s_t) \)
7. \( s_{t+1} \sim P(s_t, a_t) \)
8. \( B \leftarrow B \cup \{(s_t, a_t, r_t, s_{t+1})\} \)
9. **end for**
10. **for** each gradient step **do**
11. \( \theta \leftarrow \theta - \lambda_\pi \hat{\nabla}_\theta J_{\text{PPO}}(\pi_\theta) \)
12. \( \phi \leftarrow \phi - \lambda_V \hat{\nabla}_\phi \mathcal{L}_{\text{AC}}(\phi) \)
13. \( \phi \leftarrow \phi - \lambda_V \hat{\nabla}_\phi \mathcal{L}_{\text{AVEC}}(\phi) \)
14. **end for**
15. **end for**
Before we go further...

Do you have questions?
Continuous Control

On average, using AVEC improves SAC by 26% and PPO by 40%.

Figure: AVEC coupled with SAC and PPO on MuJoCo tasks. X-axis: number of timesteps. Y-axis: average total reward.
Sparse Reward tasks

Figure: Comparative evaluation of AVEC in sparse reward tasks. X-axis: number of timesteps. Y-axis: average total reward.

Figure: Left: State visitation frequency. Right: Phase portrait of visited states of AVEC-TRPO (green) and TRPO (red) in MountainCar.

→ AVEC is able to pick up on experienced positive reward more easily.
→ Suggests that the reconstructed shape of the value function is more accurate around such rewarding states: pushes the agent to explore further around experienced states with high values.
Does it really work? Let’s examine the new value function

1. What is the estimation error? (learning the empirical target $\hat{V}^\pi$)
2. What is the approximation error? (learning the true target $V^\pi$)
3. What is the resulting empirical variance?
Does it really work? Let’s examine the new value function

1. What is the **estimation error**? (learning the empirical target)

![Graphs showing L2 distance to \( \hat{V}^\pi \).](image)

→ **Expected result since** vanilla PPO optimizes the MSE directly.
Does it really work? Let’s examine the new value function

2. What is the **approximation error**? (learning the true target)

NB: to approximate the true value function, we fit the returns sampled from the current policy using a large number of transitions ($3 \cdot 10^5$).

![Graph showing L2 distance to $V^\pi$. X-axis: we run the algorithm and $\forall t \in \{1, 2, 4, 6, 9\} \cdot 10^5$ we stop training, use the current policy to collect $3 \cdot 10^5$ transitions to estimate $V^\pi$.]

→ **Our estimator is far closer to the true value function** half of the time (horizon is $10^6$) than the estimator obtained with MSE, then as close to it.
Does it really work? Let’s examine the new value function

3. What is the resulting **empirical variance**?

![Graph of average pairwise cosine similarity](image)

**Figure:** Average pairwise cosine similarity.

Higher average (10 batches per iteration) pairwise cosine sim. $\implies$ Closer batch-estimates of the gradient $\implies$ **Smaller gradient variance**.
Conclusion and perspectives

This paper ...:

- introduces a modification in the training objective for the critic in actor-critic algorithms which (a) is well-motivated by recent analysis of deep PG algorithms (b) produces considerable gains in performance

- highlights the benefits of a more thorough analysis of the critic objective in policy gradient methods

- is not a claim that the residual variance is the optimal loss for the state-value or the state-action-value functions, there might be better estimators hiding in the nature!

- could be followed-up by further analysis of the bias-variance trade-off (ablation study indicates that AVEC works best with $\alpha = 0$ in $\mathcal{L}_\alpha = \text{Var} + \alpha \text{Bias}^2$)

- could be extended to stochastic environments
Thank you!

Questions?


More results: approximation error

**Figure:** $L_2$ distance to $V^\pi$. X-axis: we run the algorithm and $\forall t \in \{1, 2, 4, 6, 9\} \cdot 10^5$ we stop training, use the current policy to collect $3 \cdot 10^5$ transitions to estimate $V^\pi$. 
Figure: Average cosine similarity between gradient measurements. AVEC empirically reduces the variance compared to PPO or PPO-nobaseline. Trajectory size used in estimation of the gradient variance: 3000 (upper row), 6000 (middle row), 9000 (lower row).
More results: ablation study

Figure: Sensitivity of AVEC-PPO with respect to (a,b): the bias; (c,d): the variance. X-axis: number of timesteps. Y-axis: average total reward.