Taylor Expansion Policy Optimization

Yunhao Tang [1] **Michal Valko** [2] **Rémi Munos** [2]

Abstract

In this work, we investigate the application of Taylor expansions in reinforcement learning. In particular, we propose *Taylor expansion policy optimization*, a policy optimization formalism that generalizes prior work (e.g., TRPO) as a first-order special case. We also show that Taylor expansions intimately relate to off-policy evaluation. Finally, we show that this new formulation entails modifications which improve the performance of several state-of-the-art distributed algorithms.

1. Introduction

Policy optimization is a major framework in model-free reinforcement learning (RL), with successful applications in challenging domains (Silver et al., 2016; Berner et al., 2019; Vinyals et al., 2019). Along with scaling up to powerful computational architectures (Mnih et al., 2016; Espeholt et al., 2018), significant algorithmic performance gains are driven by insights into the drawbacks of naïve policy gradient algorithms (Sutton et al., 2000). Among all algorithmic improvements, two of the most prominent are: *trust-region policy search* (Schulman et al., 2015; 2017; Abdolmaleki et al., 2018; Song et al., 2020) and *off-policy corrections* (Munos et al., 2016; Wang et al., 2017; Gruslys et al., 2018; Espeholt et al., 2018).

At the first glance, these two streams of ideas focus on orthogonal aspects of policy optimization. For trust-region policy search, the idea is to constrain the size of policy updates. This limits the deviations between consecutive policies and lower-bounds the performance of the new policy (Kakade and Langford, 2002; Schulman et al., 2015). On the other hand, off-policy corrections require that we account for the discrepancy between target policy and behavior policy. Espeholt et al. (2018) has observed that the corrections are especially useful for distributed algorithms, where behavior policy and target policy typically differ. Both algorithmic ideas have contributed significantly to stabilizing policy optimization.

In this work, we partially unify both algorithmic ideas into a single framework. In particular, we noticed that as a ubiquitous approximation method, *Taylor expansions* share high-level similarities with both trust region policy search and off-policy corrections. To get high-level intuitions of such similarities, consider a simple 1D example of Taylor expansions. Given a sufficiently smooth real-valued function on the real line $f : \mathbb{R} \rightarrow \mathbb{R}$, the k-th order Taylor expansion of $f(x)$ at x_0 is $f_k(x) \triangleq f(x_0) + \sum_{i=1}^{k} [f^{(i)}(x_0)/i!](x-x_0)^i$, where $f^{(i)}(x_0)$ are the i-th order derivatives at x_0. First, a common feature shared by Taylor expansions and trust-region policy search is the inherent notion of a trust region constraint. Indeed, in order for convergence to take place, a *trust-region constraint* is required $|x - x_0| < R(f, x_0)$[1]. Second, when using the truncation as an approximation to the original function $f_K(x) \approx f(x)$, Taylor expansions satisfy the requirement of off-policy evaluations: evaluate target policy with behavior data. Indeed, to evaluate the truncation $f_K(x)$ at any x (*target policy*), we only require the *behavior policy* “data” at x_0 (i.e., derivatives $f^{(i)}(x_0)$).

Our paper proceeds as follows. In Section 2, we start with a general result of applying Taylor expansions to Q-functions. When we apply the same technique to the RL objective, we reuse the general result and derive a higher-order policy optimization objective. This leads to Section 3, where we formally present the *Taylor Expansion Policy Optimization* (TayPO) and generalize prior work (Schulman et al., 2015; 2017) as a first-order special case. In Section , we make clear connection between Taylor expansions and $Q(\lambda)$ (Harutyunyan et al., 2016), a common return-based off-policy evaluation operator. Finally, in Section 5, we show the performance gains due to the higher-order objectives across a range of state-of-the-art distributed deep RL agents.

2. Taylor expansion for reinforcement learning

Consider a Markov Decision Process (MDP) with state space $\mathcal{X}$ and action space $\mathcal{A}$. Let policy $\pi(\cdot|x)$ be a dis-

[1]Columbia University, New York, USA [2]DeepMind Paris, France. Correspondence to: yt2541@coluymbia.edu <Yunhao>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020. Copyright 2020 by the author(s).

[1] Here, $R(f, x_0)$ is the convergence radius of the expansions, which in general depends on the function f and origin x_0.

tribution over actions give state x. At a discrete time $t \geq 0$, the agent in state x_t takes action $a_t \sim \pi(\cdot|x_t)$, receives reward $r_t \triangleq r(x_t, a_t)$, and transitions to a next state $x_{t+1} \sim p(\cdot|x_t, a_t)$. We assume a discount factor $\gamma \in [0, 1)$. Let $Q^\pi(x, a)$ be the action value function (Q-function) from state x, taking action a, and following policy π. For convenience, we use $d_\gamma^\pi(\cdot, \cdot|x_0, a_0, \tau)$ to denote the discounted visitation distribution starting from state-action pair (x_0, a_0) and following π, such that $d_\gamma^\pi(x, a|x_0, a_0, \tau) = (1-\gamma)\gamma^{-\tau} \sum_{t \geq \tau} \gamma^t P(x_t = x|x_0, a_0, \pi)\pi(a|x)$. We thus have $Q^\pi(x, a) = (1-\gamma)^{-1}\mathbb{E}_{(x',a') \sim d_\gamma^\pi(\cdot,\cdot|x,a,0)}[r(x', a')]$. We focus on the RL objective of optimizing $\max_\pi J(\pi) \triangleq \mathbb{E}_{\pi, x_0}[\sum_{t \geq 0} \gamma^t r_t]$ starting from a fixed initial state x_0.

We define some useful matrix notation. For ease of analysis, we assume that $\mathcal{X}$ and $\mathcal{A}$ are both finite. Let $R \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{A}|}$ denote the reward function and $P^\pi \in \mathbb{R}^{|\mathcal{X}||\mathcal{A}| \times |\mathcal{X}||\mathcal{A}|}$ denote the transition matrix such that $P^\pi(x, a, y, b) \triangleq p(y|x, a)\pi(b|y)$. We also define $Q^\pi \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{A}|}$ as the vector Q-function. This matrix notation facilitates compact derivations, for example, the Bellman equation writes as $Q^\pi = R + \gamma P^\pi Q^\pi$.

2.1. Taylor Expansion of Q-functions.

In this part, we state the Taylor expansion of Q-functions. Our motivation for the expansion is the following: Assume we aim to estimate $Q^\pi(x, a)$ for target policy π, and we only have access to data collected under a behavior policy μ. Since $Q^\mu(x, a)$ can be readily estimated with the collected data, how do we approximate $Q^\pi(x, a)$ with $Q^\mu(x, a)$?

Clearly, when $\pi = \mu$, then $Q^\pi = Q^\mu$. Whenever $\pi \neq \mu$, Q^π starts to deviate from Q^μ. Therefore, we apply Taylor expansion to describe the deviation $Q^\pi - Q^\mu$ in the orders of $P^\pi - P^\mu$. We provide the following result.

Theorem 1. *(proved in Appendix B) For any policies π and μ, and any $K \geq 1$, we have*

$$Q^\pi - Q^\mu = \sum_{k=1}^{K} \big(\gamma(I - \gamma P^\mu)^{-1}(P^\pi - P^\mu)\big)^k Q^\mu + \big(\gamma(I - \gamma P^\mu)^{-1}(P^\pi - P^\mu)\big)^{K+1} Q^\pi.$$

In addition, if $||\pi - \mu||_1 \triangleq \max_x \sum_a |\pi(a|x) - \mu(a|x)| < (1-\gamma)/\gamma$, then the limit for $K \to \infty$ exists and we have

$$Q^\pi - Q^\mu = \sum_{k=1}^{\infty} \underbrace{\big(\gamma(I - \gamma P^\mu)^{-1}(P^\pi - P^\mu)\big)^k Q^\mu}_{\triangleq U_k}. \quad (1)$$

The constraint between π and μ is a result of the convergence radius of the Taylor expansion. The derivation follows by recursively applying the following equality: $Q^\pi = Q^\mu + \gamma(I - \gamma P^\mu)^{-1}(P^\pi - P^\mu)Q^\pi$. Please refer to the Appendix B for a proof. For ease of notation, denote the k-th term on the RHS of Eq. 1 as U_k. This gives rise to $Q^\pi - Q^\mu = \sum_{k=1}^\infty U_k$.

To represent $Q^\pi - Q^\mu$ explicitly with the deviation between π and μ, consider a diagonal matrix $D_{\pi/\mu}(x, a, y, b) \triangleq \pi(a|x)/\mu(b|y) \cdot \delta_{x=y,a=b}$ where $x, y \in \mathcal{X}, a, b \in \mathcal{A}$ and where δ is the Dirac delta function; we restrict to the case where $\mu(a|x) > 0, \forall x, a$. This diagonal matrix $D_{\pi/\mu} - I$ is a measure of the deviation between π and μ. The above expression can be rewritten as

$$Q^\pi - Q^\mu = \sum_{k=1}^{\infty} (\gamma(I - \gamma P^\mu)^{-1} P^\mu (D_{\pi/\mu} - I))^k Q^\mu. \quad (2)$$

We will see that the expansion in Eq. 2 is useful in Section3 when we derive the Taylor expansion of the difference between the performances of two policies, $J(\pi) - J(\mu)$. In Section 4, we also provide the connection between Taylor expansion and off-policy evaluation.

2.2. Taylor expansion of reinforcement learning objective

When searching for a better policy, we are often interested in the difference $J(\pi) - J(\mu)$. With Eq. 2, we can derive a similar Taylor expansion result for $J(\pi) - J(\mu)$. Let π_t (resp., μ_t) be the shorthand notation for $\pi(a_t|x_t)$ (resp., $\mu(a_t|x_t)$). Here, we formalize the orders of the expansion as the number of times that ratios $\pi_t/\mu_t - 1$ appear in the expression, e.g., the first-order expansion should only involve $\pi_t/\mu_t - 1$ up to the first order, without higher order terms, e.g., cross product $(\pi_t/\mu_t - 1)(\pi_{t'}/\mu_{t'} - 1)$. We denote the k-th order as $L_k(\pi, \mu)$ and by construction $J(\pi) - J(\mu) = \sum_{k=1}^\infty L_k(\pi, \mu)$. Next, we derive practically useful expressions for $L_k(\pi, \mu)$.

We provide a derivation sketch below and give the details in Appendix F. Let $\pi_0, \mu_0 \in \mathbb{R}^{|\mathcal{X}| \times |A|}$ be the joint distribution of policies and state at time $t = 0$ such that $\pi_0(x, a) = \pi(a|x)\delta_{x=x_0}$. Note that the RL objective equivalently writes as $J(\pi) = V^\pi(x_0) = \sum_a \pi(a|x_0)Q^\pi(x_0, a)$ and can be expressed as an inner product $J(\pi) = \pi_0^\intercal Q^\pi$. This allows us to import results from Eq. 2,

$$\begin{aligned} J(\pi) - J(\mu) &= \pi_0^\intercal Q^\pi - \mu_0^\intercal Q^\mu \quad (3) \\ &= (\pi_0 - \mu_0)^\intercal \left(Q^\mu + \sum_{k \geq 1} U_k \right) + \mu_0^\intercal \left(\sum_{k \geq 1} U_k \right). \end{aligned}$$

By reading off different orders of the expansion from the RHS of Eq. 3, we derive

$$\begin{aligned} L_1(\pi, \mu) &= (\pi_0 - \mu_0)^\intercal Q^\mu + \mu_0^\intercal U_1, \quad (4) \\ L_k(\pi, \mu) &= (\pi_0 - \mu_0)^\intercal U_{k-1} + \mu_0^\intercal U_k, \ \forall k \geq 2. \end{aligned}$$

It is worth noting that the k-th order expansion of the RL objective $L_k(\pi,\mu)$ is a mixture of the $(k-1)$-th and k-th order Q-function expansions. This is because $J(\pi)$ integrates Q^π over the initial π_0 and the initial difference $\pi_0-\mu_0$ contributes one order of difference in $L_k(\pi,\mu)$.

Below, we illustrate the results for $k=1,2$, and $k\geq 3$. To make the results more intuitive, we convert the matrix notation of Eq. 3 into explicit expectations under μ.

First-order expansion. By converting $L_1(\pi,\mu)$ from Eq. 4 into expectations, we get

$$\mathop{\mathbb{E}}_{\substack{x,a\sim d_\gamma^\mu(\cdot,\cdot|x_0,a_0,0),\\ a_0\sim\mu(\cdot|x_0)}}\left[\left(\frac{\pi(a|x)}{\mu(a|x)}-1\right)Q^\mu(x,a)\right]. \tag{5}$$

To be precise $L_1(\pi,\mu)=(1-\gamma)^{-1}\times$ (Eq. 5) to account for the normalization of the distribution d_γ^μ. Note that $L_1(\pi,\mu)$ is exactly the same as surrogate objective proposed in prior work on scalable policy optimization (Kakade and Langford, 2002; Schulman et al., 2015; 2017). Indeed, these works proposed to estimate and optimize such a surrogate objective at each iteration while enforcing a trust region. In the following, we generalize this objective with Taylor expansions.

Second-order expansion. By converting $L_2(\pi,\mu)$ from Eq. 4 into expectations, we get

$$\mathop{\mathbb{E}}_{\substack{x,a\sim d_\gamma^\mu(\cdot,\cdot|x_0,a_0,0),\\ a_0\sim\mu(\cdot|x_0)\\ x',a'\sim d_\gamma^\mu(\cdot,\cdot|x,a,1)}}\left[\left(\frac{\pi(a|x)}{\mu(a|x)}-1\right)\left(\frac{\pi(a'|x')}{\mu(a'|x')}-1\right)Q^\mu(x',a')\right]. \tag{6}$$

Again, accounting for the normalization, $L_2(\pi,\mu)=\gamma(1-\gamma)^{-2}\times$(Eq. 6). To calculate the above expectation, we first start from (x_0,a_0), and sample a pair (x,a) from the discounted distribution $d_\gamma^\mu(\cdot,\cdot|x_0,a_0,0)$. Then, we use (x,a) as the starting point and sample another pair from $d_\gamma^\mu(\cdot,\cdot|x,a,1)$. This implies that the second-order expansion can be estimated only via samples under μ, which will be essential for policy optimization in practice.

It is worth noting that the second state-action pair $(x',a')\sim d_\gamma^\mu(\cdot,\cdot|x,a,1)$ with the argument $\tau=1$ instead of $\tau=0$. This is because $L_k(\pi,\mu),k\geq 2$ only contains terms π_t/μ_t-1 sampled across strictly different time steps.

Higher-order expansions. Similarly to the first-order and second-order expansions, higher-order expansions are also possible by including proper higher-order terms in π_t/μ_t-1. For general $K\geq 1$, $L_K(\pi,\mu)$ can be expressed as (omitting the normalization constants)

$$\mathbb{E}_{(x^{(i)},a^{(i)})_{1\leq i\leq K}}\left[\prod_{i=1}^{K}\left(\frac{\pi(a^{(i)}|x^{(i)})}{\mu(a^{(i)}|x^{(i)})}-1\right)Q^\mu(x^{(K)},a^{(K)})\right]. \tag{7}$$

Here, $(x^{(i)},a^{(i)}),1\leq i\leq K$ are sampled sequentially, each following a discounted visitation distribution conditional on the previous state-action pair. We show their detailed derivations in Appendix F. Furthermore, we discuss the trade-off of different orders K in Section 3.

Interpretation & intuition. Evaluating $J(\pi)$ with data under μ requires importance sampling (IS) $J(\pi)=\mathbb{E}_{\mu,x_0}[(\Pi_{t\geq 0}\frac{\pi_t}{\mu_t})(\sum_{t\geq 0}\gamma^t r_t)]$. In general, since π can differ from μ at all $|\mathcal{X}||\mathcal{A}|$ state-action pairs, computing $J(\pi)$ exactly with full IS requires corrections at all steps along generated trajectories. First-order expansion (Eq. 5) corresponds to carrying out only one single correction at sampled state-action pair along the trajectories: Indeed, in computing Eq. 5, we sample a state-action pair (x,a) along the trajectory and calculate one single IS correction $(\pi(a|x)/\mu(a|x)-1)$. Similarly, the second-order expansion (Eq. 6) goes one step further and considers the IS correction at two different steps (x,a) and (x',a'). As such, Taylor expansions of the RL objective can be interpreted as increasingly tight approximations of the full IS correction.

3. Taylor expansion for policy optimization

In high-dimensional policy optimization, where exact algorithms such as dynamic programming are not feasible, it is necessary to learn from sampled data. In general, the sampled data are collected under a behavior policy μ different from the target policy π. For example, in trust-region policy search (e.g., TRPO, Schulman et al., 2015; PPO, Schulman et al., 2017), π is the new policy while μ is a previous policy; in asynchronous distributed algorithms (Mnih et al., 2016; Espeholt et al., 2018; Horgan et al., 2018; Kapturowski et al., 2019), π is the learner policy while μ is delayed actor policy. In this section, we show the fundamental connection between trust-region policy search and Taylor expansions, and propose the general framework of *Taylor expansion policy optimization* (TayPO).

3.1. Generalized trust-region policy optimization

For policy optimization, it is necessary that the update function (e.g., policy gradients or surrogate objectives) can be estimated with sampled data under behavior policy μ. Taylor expansions are a natural paradigm to satisfy this requirement. Indeed, to optimize $J(\pi)$, consider optimizing[2]

$$\max_\pi \ J(\pi) = \max_\pi \ J(\mu) + \sum_{k=1}^{\infty} L_k(\pi, \mu). \tag{8}$$

Though we have shown that for all k, $L_k(\pi, \mu)$ are expectations under μ, it is not feasible to unbiasedly estimate the RHS of Eq. 8 because it involves an infinite number of terms. In practice, we can truncate the objective up to K-th order $\sum_{k=1}^K L_k(\pi, \mu)$ and drop $J(\mu)$ because it does not involve π.

However, for any fixed K, optimizing the truncated objective $\sum_{k=1}^K L_k(\pi, \mu)$ in an unconstrained way is risky: As π, μ become increasingly different, the approximation $J(\mu) + \sum_{k=1}^K L_k(\pi, \mu) \approx J(\pi)$ becomes more inaccurate and we stray away from optimizing $J(\pi)$, the objective of interest. The approximation error comes from the residual $E_K \triangleq \sum_{k=K+1}^{\infty} U_k$ — to control the magnitude of the residual, it is natural to constrain $||\pi - \mu||_1 \leq \varepsilon$ with some $\varepsilon > 0$. Indeed, it is straightforward to show that

$$||E_K||_\infty \leq \left(\frac{\gamma\varepsilon}{1-\gamma}\right)^{K+1} \left(1 - \frac{\gamma\varepsilon}{1-\gamma}\right)^{-1} \frac{R_{\text{max}}}{1-\gamma},$$

where $R_{\text{max}} \triangleq \max_{x,a} |r(x, a)|$.[3] Please see Appendix A.1 for more detailed derivations. We formalize the entire local optimization problem as *generalized trust-region policy optimization* (generalized TRPO),

$$\max_\pi \sum_{k=1}^{K} L_k(\pi, \mu), \ \ ||\pi - \mu||_1 \leq \varepsilon. \tag{9}$$

Monotonic improvement. While maximizing the surrogate objective under trust-region constraints (Eq. 9), it is desirable to have performance guarantee on the true objective $J(\pi)$. Below, Theorem 2 gives such a result.

Theorem 2. *(proved in Appendix C) When the policy π is optimized based on the trust-region objective Eq. 9 and $\varepsilon < \frac{1-\gamma}{\gamma}$, the performance $J(\pi)$ is lower bounded as*

$$J(\pi) \geq J(\mu) + \sum_{k=1}^{K} L_k - G_K, \tag{10}$$

where $G_K \triangleq \dfrac{1}{\gamma(1-\gamma)}\left(1 - \dfrac{\gamma}{1-\gamma}\varepsilon\right)^{-1} \left(\dfrac{\gamma\varepsilon}{1-\gamma}\right)^{K+1} R_{max}.$

Note that if $\varepsilon < (1-\gamma)/\gamma$, then as $K \to \infty$, the gap $G_K \to 0$. Therefore, when optimizing π based on Eq. 9, the performance $J(\pi)$ is always lower-bounded according to Eq. 10.

Connections to prior work on trust-region policy search. The generalized TRPO extends the formulation of prior work, e.g., TRPO/PPO of Schulman et al. (2015; 2017). Indeed, idealized forms of these algorithms are a special case for $K = 1$, though for practical purposes the ℓ_1 constraint is replaced by averaged KL constraints.[4]

3.2. TayPO-k: Optimizing with k-th order expansion

Though there is a theoretical motivation to use trust-region constraints for policy optimization (Schulman et al., 2015; Abdolmaleki et al., 2018), such constraints are rarely explicitly enforced in practice in its most standard form (Eq. 9). Instead, trust regions are *implicitly encouraged* via e.g., ratio clipping (Schulman et al., 2017) or parameter averaging (Wang et al., 2017). In large-scale distributed settings, algorithms already benefit from diverse sample collections for variance reduction of the parameter updates (Mnih et al., 2016; Espeholt et al., 2018), which brings the desired stability for learning and makes trust-region constraints less necessary (either explicit or implicit). Therefore, we focus on the setting where no trust region is explicitly enforced. We introduce a new family of algorithm **TayPO-k**, which applies the k-th order Taylor expansions for policy optimization.

Unbiased estimations with variance reduction. In practice, $L_k(\pi_\theta, \mu)$ as expectations under μ can be estimated as $\hat{L}_k(\pi_\theta, \mu)$ over a single trajectory. Take $K = 2$ as an example: Given a trajectory $(x_t, a_t, r_t)_{t=0}^{\infty}$ by μ, assume we have access to some estimates of $Q^\mu(x, a)$, e.g., cumulative returns. To generate a sample from $(x, a) \sim d_\gamma^\mu(x_0, a_0, 0)$, we can first sample a random time from a geometric distribution with success probability $1 - \gamma$, i.e., $t \sim \text{Geometric}(1 - \gamma)$. Second, we sample another random time t' with geometric distribution $\text{Geometric}(1 - \gamma)$ but conditional on $t' \geq 1$.[5] Then, a single sample estimate of Eq. 6 is given by

$$\left(\frac{\pi(a_t|x_t)}{\mu(a_t|x_t)} - 1\right)\left(\frac{\pi(a_{t+t'}|x_{t+t'})}{\mu(a_{t+t'}|x_{t+t\prime})} - 1\right) Q^\mu(x_{t+t'}, a_{t+t'}).$$

Further, the following shows the effect of replacing Q-values $Q^\mu(x, a)$ by advantages $A^\mu(x, a) \triangleq Q^\mu(x, a) - V^\mu(x)$.

Theorem 3. *(proved in Appendix D) The computation of $L_k(\pi, \mu)$ based on Eq. 7 is exact when replacing $Q^\mu(x, a)$ by $A^\mu(x, a)$, i.e. $L_k(\pi, \mu), k \geq 1$ can be expressed as*

$$\mathbb{E}_{(x^{(i)}, a^{(i)})_{1 \leq i \leq K}} \left[\prod_{i=1}^{K} \left(\frac{\pi(a^{(i)}|x^{(i)})}{\mu(a^{(i)}|x^{(i)})} - 1\right) A^\mu(x^{(K)}, a^{(K)})\right].$$

[2] Once again, the equality $J(\pi) = J(\mu) + \sum_{k=1}^{\infty} L_k(\pi, \mu)$ holds under certain conditions, detailed in Section 4.

[3] Here we define $||E||_\infty \triangleq \max_{x,a} |E(x, a)|$.

[4] Instead of forming the constraints explicitly, PPO (Schulman et al., 2017) enforces the constraints implicitly by clipping IS ratios.

[5] As explained in Section 2.2, since $L_2(\pi, \mu)$ contains IS ratios at strictly different time steps, it is required that $t' \geq 1$.

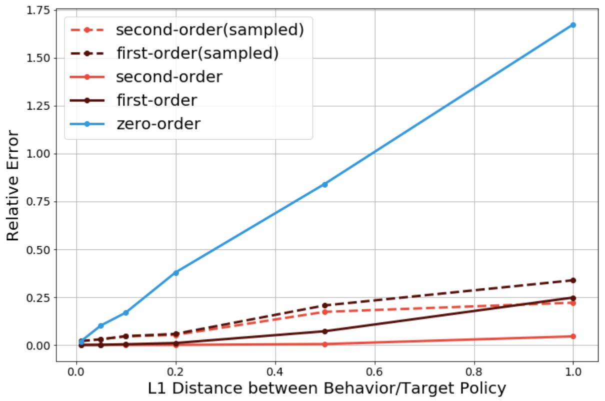

Figure 1. Experiments on a small MDP. The x-axis measures $|\pi - \mu|_1$ and the y-axis shows the relative errors in off-policy estimates. All errors are computed analytically. Solid lines are computed with ground-truth rewards R while dashed lines with estimates $\hat{R}$.

In practice, when computing $\hat{L}_k(\pi, \mu)$, replacing $\hat{Q}^\mu(x, a)$ by $\hat{A}^\mu(x, a)$ still produces an unbiased estimate *and* potentially reduces variance. This naturally recovers the result in prior work for $K = 1$ (Schulman et al., 2016).

Higher-order objectives and trade-offs. When $K \geq 3$, we can construct objectives with higher-order terms. The motivation is that with high K, $\sum_{k=1}^{K} L_k(\pi_\theta, \mu)$ forms a closer approximation to the objective of interest: $J(\pi) - J(\mu)$. Why not then have K as large as possible? This comes at a trade-off. For example, let us compare $L_1(\pi_\theta, \mu)$ and $L_1(\pi_\theta, \mu) + L_2(\pi_\theta, \mu)$: Though $L_1(\pi_\theta, \mu) + L_2(\pi_\theta, \mu)$ forms a closer approximation to $J(\pi) - J(\mu)$ than $L_1(\pi)$ *in expectation*, it could have higher variance during estimation when e.g., $L_1(\pi_\theta, \mu)$ and $L_2(\pi_\theta, \mu)$ have a non-negative correlation. Indeed, as $K \to \infty$, $\sum_{k=1}^{K} L_k(\pi_\theta, \mu)$ approximates the full IS correction, which is known to have high variance (Munos et al., 2016).

How many orders to take in practice? Though the higher-order policy optimization formulation generalizes previous results (Schulman et al., 2015; 2017) as an first-order special case, does it suffice to only include first-order terms in practice?

To assess the effects of Taylor expansions, consider a policy evaluation problem on a random MDP (see Appendix H.1 for the detailed setup): Given a target policy π and a behavior policy μ, the approximation error of the K-th order expansion is $e_K \triangleq Q^\pi - (Q^\mu + \sum_{k=1}^{K} U_k)$. In Figure 1, We show the relative errors $||e_K||_1 / ||Q^\pi||_1$ as a function of $\varepsilon = ||\pi - \mu||_1$. Ground-truth quantities such as Q^π are always computed analytically. Solid lines show results where all estimates are also computed analytically, e.g., Q^μ is computed as $(I - \gamma P^\mu)^{-1} R$. Observe that the errors decrease drastically as the expansion order $K \in \{0, 1, 2\}$ increases. To quantify how sample estimates impact the quality of approximations, we re-compute the estimates but with R replaced by empirical estimates $\hat{R}$. Results are shown in dashed curves. Now comparing $K = 1, 2$, observe that both errors go up compared to their fully analytic counterparts - both become more similar when ε is small.

This provides motivations for second-order expansions. While first-orders are a default choice for common deep RL algorithms (Schulman et al., 2015; 2017), from the simple MDP example we see that the second-order expansions could potentially improve upon the first-order, *even with sample estimates*.

Algorithm 1 TayPO-2: Second-order policy optimization

```
Require: policy $\pi_\theta$ with parameter $\theta$ and $\alpha, \eta > 0$
  while not converged do
    1. Collect partial trajectories $(x_t, a_t, r_t)_{t=1}^{T}$ under behavior policy $\mu$.
    2. Estimate on-policy advantage from the trajectories $\hat{A}^\mu(x_t, a_t)$.
    3. Construct first-order/second-order surrogate objective function $\hat{L}_1(\pi_\theta, \mu), \hat{L}_2(\pi_\theta, \mu)$ according to Eq. 5, Eq. 6 respectively, replacing $Q^\mu(x, a)$ by $\hat{A}^\mu(x, a)$.
    4. The full objective $\hat{L}_\theta \leftarrow \hat{L}_1(\pi_\theta, \mu) + \hat{L}_2(\pi_\theta, \mu)$.
    5. Gradient update $\theta \leftarrow \theta + \alpha \nabla_\theta \hat{L}_\theta$.
  end while
```

3.3. TayPO-2 — Second-order policy optimization

From here onwards, we focus on TayPO-2. At any iteration, the data are collected under behavior policy μ in the form of partial trajectories $(x_t, a_t, r_t)_{t=1}^{T}$ of length T. The learner maintains a parametric policy π_θ to be optimized. First, we carry out advantage estimation $\hat{A}^\mu(x, a)$ for state-action pairs on the partial trajectories. This could be naïvely estimated as $\hat{A}^\mu(x_t, a_t) = \sum_{t' \geq t}^{T-1} r_{t'} \gamma^{t'-t} + V_\varphi(x_T)\gamma^{T-t} - V_\varphi(x_t)$ where $V_\varphi(x)$ are value function baselines. One could also adopt more advanced estimation techniques such as *generalized advantage estimation* (GAE, Schulman et al., 2016). Then, we construct surrogate objectives for optimization: the first-order component $\hat{L}_1(\pi_\theta, \mu)$ as well as second-order component $\hat{L}_2(\pi_\theta, \mu) - \hat{L}_1(\pi_\theta, \mu)$, based on Eq. 5 and Eq. 6 respectively. Note that we replace all $Q^\mu(x, a)$ by $\hat{A}^\mu(x, a)$ for variance reduction.

Therefore, our final objective function becomes

$$\hat{L}_\theta \triangleq \hat{L}_1(\pi_\theta, \mu) + \hat{L}_2(\pi_\theta, \mu). \qquad (11)$$

The parameter is updated via gradient ascent $\theta \leftarrow \theta + \alpha \nabla \hat{L}_\theta$. Similar ideas can be applied to *value-based algorithms*, for which we provide details in Appendix G.

4. Unifying the concepts: Taylor expansion as return-based off-policy evaluation

So far we have made the connection between Taylor expansions and TRPO. On the other hand, as introduced in Section 1, Taylor expansions can also be intimately related to *off-policy evaluation*. Below, we formalize their connections. With Taylor expansions, we provide a consistent and unified view of TRPO and off-policy evaluation.

4.1. Taylor expansion as off-policy evaluation

In the general setting of off-policy evaluation, the data is collected under a behavior policy μ while the objective is to evaluate Q^π. *Return-based off-policy evaluation operators* (Munos et al., 2016) are a family of operators $\mathcal{R}_c^{\pi,\mu} : \mathbb{R}^{|\mathcal{X}||\mathcal{A}|} \mapsto \mathbb{R}^{|\mathcal{X}||\mathcal{A}|}$, indexed by (per state-action) trace-cutting coefficients $c(x,a)$, a behavior policy μ and a target policy π,

$$\mathcal{R}_c^{\pi,\mu} Q \triangleq Q + (I - \gamma P^{c\mu})^{-1}(r + \gamma P^\pi Q - Q),$$

where $P^{c\mu}$ is the *(sub)-probability transition kernel* for policy $c(x',a')\mu(a'|x')$. Starting from any Q-function Q, repeated applications of the operator will result in convergence to Q^π, i.e.,

$$(\mathcal{R}_c^{\pi,\mu})^K Q \to Q^\pi,$$

as $K \to \infty$, subject to certain conditions on $c(x,a)$. To state the main results, recall that Eq. 2 rewrites as $Q^\pi = \lim_{K\to\infty}\left(Q^\mu + \sum_{k=1}^K U_k\right)$. In practice, we take a finite K and use the approximation $Q^\mu + \sum_{k=1}^K U_k \approx Q^\pi$.

Next, we state the following result establishing a connection between K-th order Taylor expansion and the return-based off-policy operator applied K times.

Theorem 4. *(proved in Appendix E) For any $K \geq 1$, any policies π and μ,*

$$Q^\mu + \sum_{k=1}^K U_k = (\mathcal{R}_1^{\pi,\mu})^K Q^\mu, \tag{12}$$

where $\mathcal{R}_1^{\pi,\mu}$ is short for $c(x,a) \equiv 1$.

Theorem 4 shows that when we approximate Q^π by the Taylor expansion up to the K-th order, $Q^\mu + \sum_{k=1}^K U_k$, it is equivalent to generating an approximation by K times applying the off-policy evaluation operator $\mathcal{R}_1^{\pi,\mu}$ on Q^μ. We also note that the off-policy evaluation operator in Theorem 4 is the $Q(\lambda)$ operator (Harutyunyan et al., 2016) with $\lambda = 1$.[6]

Alternative proof for $Q(\lambda)$ convergence for $\lambda = 1$. Since Taylor expansions converge within a convergence radius, which in this case corresponds to $||\pi - \mu||_1 < (1-\gamma)/\gamma$, it implies that $Q(\lambda)$ with $\lambda = 1$ converges when this condition holds. In fact, this coincides with the condition deduced by Harutyunyan et al. (2016).[7]

4.2. An operator view of trust-region policy optimization

With the connection between Taylor expansion and off-policy evaluation, along with the connection between Taylor expansion and TRPO (Section 3) we give a novel interpretation of TRPO: The K-th order generalized TRPO is approximately equivalent to iterating K times the off-policy evaluation operator $\mathcal{R}_1^{\pi,\mu}$.

To make our claim explicit, recall the RL objective in matrix form is $J(\pi) = \pi_0^\intercal Q^\pi$. Now consider approximating Q^π by applying the evaluation operator $\mathcal{R}_1^{\pi,\mu}$ to Q^μ, iterating K times. This produces the surrogate objective $\pi_0^\intercal (\mathcal{R}_1^{\pi,\mu})^K Q^\mu \approx J(\mu) + \sum_{k=1}^K L_k(\pi,\mu)$, approximately equivalent to that of the generalized TRPO (Eq. 9).[8] As a result, the generalized TRPO (including TRPO; Schulman et al., 2015) can be interpreted as approximating the exact RL objective $J(\pi)$, by K times iterating the evaluation operator $\mathcal{R}_1^{\pi,\mu}$ on Q^μ to approximate Q^π. When does this evaluation operator converge? Recall that $\mathcal{R}_1^{\pi,\mu}$ converges when $||\pi - \mu||_1 < (1-\gamma)/\gamma$, i.e., there is a trust region constraint on π, μ. This is consistent with the motivation of generalized TRPO discussed in Section 3, where a trust region is required for monotonic improvements.

5. Experiments

We evaluate the potential benefits of applying second-order expansions in a diverse set of scenarios. In particular, we test if the second-order correction helps with **(1)** *policy-based* and **(2)** *value-based* algorithms.

In large-scale experiments, to take advantage of computational architectures, actors (μ) and learners (π) are not perfectly synchronized. For case **(1)**, in Section 5.1, we show that even in cases where they almost synchronize ($\pi \approx \mu$), higher-order corrections are still helpful. Then, in Section 5.2, we study how the performance of a general distributed policy-based agent (e.g., IMPALA, Espeholt et al., 2018) is influenced by the discrepancy between actors and learners. For case **(2)**, in Section 5.3, we show the benefits of second-order expansions in with a state-of-the-art value-based agent

[6] As a side note, we also show that the advatnage estimation method GAE (Schulman et al., 2016) is highly related to the $Q(\lambda)$ operator in Appendix F.1.

[7] Note that this alternative proof only works for the case where the initial $Q_{\text{init}} = Q^\mu$.

[8] The k-th order Taylor expansion of Q^π is slightly different from that of the RL objective $J(\pi)$ by construction; see Appendix B for details.

R2D2 (Kapturowski et al., 2019).

Evaluation. All evaluation environments are done on the entire suite of Atari games (Bellemare et al., 2013). We report human-normalized scores for each level, calculated as $z_i = (r_i - o_i)/(h_i - o_i)$, where h_i and o_i are the performances of human and a random policy on level i respectively; with details in Appendix H.2.

Architecture for distributed agents. Distributed agents generally consist of a central learner and multiple actors (Nair et al., 2015; Mnih et al., 2016; Babaeizadeh et al., 2017; Barth-Maron et al., 2018; Horgan et al., 2018). We focus on two main setups: **Type I** includes agents such as IMPALA (Espeholt et al., 2018) (see blue arrows in Figure 5 in Appendix H.3). See Section 5.1 and Section 5.2; **Type II** includes agents such as R2D2 (Kapturowski et al., 2019; see orange arrows in Figure 5 in Appendix H.3). See Section 5.3. We provide details on hyper-parameters of experiment setups in respective subsections in Appendix H.

Practical considerations. We can extend the TayPO-2 objective (Eq. 11) to $\hat{L}_\theta = \hat{L}_1(\pi_\theta, \mu) + \eta \hat{L}_2(\pi_\theta, \mu)$ with $\eta > 0$. By choosing η, one achieves bias-variance trade-offs of the final objective and hence the update. We found $\eta = 1$ (exact TayPO-2) working reasonably well. See Appendix H.4 for the ablation study on η and further details.

5.1. Near on-policy policy optimization

The policy-based agent maintains a target policy network $\pi = \pi_\theta$ for the learner and a set of behavior policy networks $\mu = \pi_{\theta'}$ for the actors. The actor parameters θ' are delayed copies of the learner parameter θ. To emulate a near on-policy situation $\pi \approx \mu$, we minimize the delay of the parameter passage between the central learner and actors, by hosting both learner/actors on the same machine.

We compare second-order expansions with two baselines: *first-order* and *zero-order*. For the first-order baseline, we also adopt the PPO technique of clipping: $\text{clip}(\pi(a|x)/\mu(a|x), 1-\varepsilon, 1+\varepsilon)$ in Eq. 5 with $\varepsilon = 0.2$. Clipping the ratio enforces an implicit trust region with the goal of increased stability (Schulman et al., 2017). This technique has been shown to generally outperform a naïve explicit constraint, as done in the original TRPO (Schulman et al., 2015). In Appendix H.5, we detail how we implemented PPO on the asynchronous architecture. Each baseline trains on the entire Atari suite for 400M frames and we compare the mean/median human-normalized scores.

The comparison results are shown in Figure 2. Please see the median score curves in Figure 6 in Appendix H.5. We make several observations: **(1)** Off-policy corrections are very critical. Going from zero-order (no correction) to first-order improves the performance most significantly, even when the delays between actors and the learner are minimized as much as possible; **(2)** Second-order correction significantly improves on the first-order baseline. This might be surprising, because when near on-policy, one should expect the difference between additional second-order correction to be less important. This implies that in fully asynchronous architecture, it is challenging to obtain sufficiently on-policy data and additional corrections can be helpful.

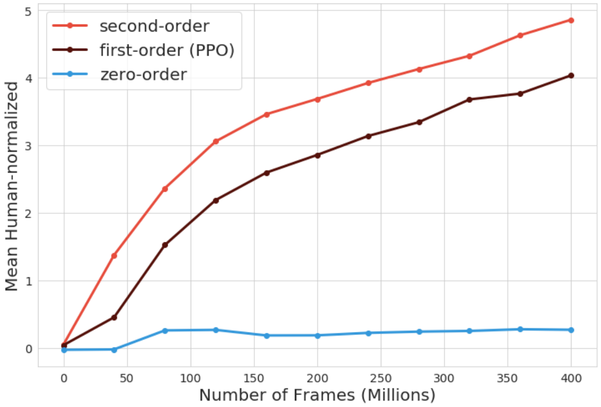

Figure 2. **Near on-policy optimization.** The x-axis is the number of frames (millions) and y-axis shows the mean human-normalized scores averaged across 57 Atari levels. The plot shows the mean curve averaged across 3 random seeds. We observe that second-order expansions allow for faster learning and better asymptotic performance given the fixed budget on actor steps.

5.2. Distributed off-policy policy optimization

We adopt the same setup as in Section 5.1. To maximize the overall throughput of the agent, the central learner and actors are distributed on different host machines. As a result, both parameter passage from the learner to actors and data passage from actors to the learner could be severely delayed. This creates a natural off-policy scenario with $\pi \neq \mu$.

We compare second-order with two baselines: *first-order* and *V-trace*. The V-trace is used in the original IMPALA agent (Espeholt et al., 2018) and we present its details in Appendix H.6. We are interested in how the agent's performance changes as the level of off-policy increases. In practice, the level of off-policy can be controlled and measured as the delay (measured in milliseconds) of the parameter passage from the learner to actors. Results are shown in Figure 3, where x-axis shows the artificial delays (in $\log$ scale) and y-axis shows the mean human-normalized scores after training for 400M frames. Note that the total delay consists of both artificial delays and inherent delays in the distributed system.

We make several observations: **(1)** All baseline variants'

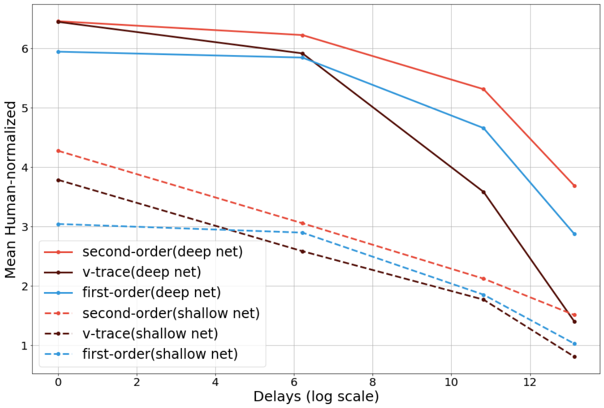

Figure 3. **Distributed off-policy policy optimization.** The x-axis is the controlled delays between the actors and learner (in log scale) and y-axis shows the mean human-normalized scores averaged across 57 Atari levels after training for 400M frames. Each curve averages across 3 random seeds. Solid curves are results trained with resnets while dashed curves are trained with shallow nets second-order expansions make little difference compared to baselines (V-trace and first-order) when the delays are small. When delays increase, the performance of second-order expansions decay more slowly.

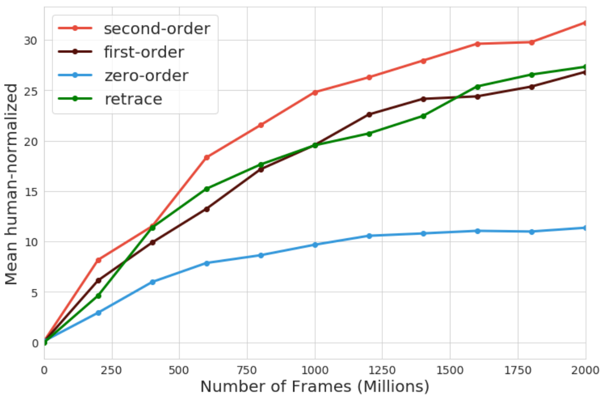

Figure 4. **Value-based learning with distributed architecture (R2D2).** The x-axis is number of frames (millions) and y-axis shows the mean human-normalized scores averaged across 57 Atari levels over the training of 2000M frames. Each curve averages across 2 random seeds. The second-order correction performs marginally better than first-order correction and retrace, and significantly better than zero-order. See Appendix G for detailed descriptions of these baseline variants.

performance degrades as the delays increase. All baseline off-policy corrections are subject to failures as the level of off-policines increases. **(2)** While all baselines perform rather similarly when delays are small, as the level of off-policy increases, second-order correction degrades slightly more gracefully than the other baselines. This implies that second-order is a more robust off-policy correction method than other current alternatives.

5.3. Distributed value-based learning

The value-based agent maintains a Q-function network Q_θ for the learner and a set of delayed Q-function networks $Q_{\theta'}$ for the actors. Let $\mathcal{E}$ be an operator such that $\mathcal{E}(Q, \varepsilon)$ returns the ε-greedy policy with respect to Q. The actors generate partial trajectories by executing an $\mu = \mathcal{E}(Q_{\theta'}, \varepsilon)$ and send data to a replay buffer. The target policy is greedy with respect to the current Q-function $\pi = \mathcal{E}(Q_\theta, 0)$. The learner samples partial trajectories from the replay buffer and updates parameters by minimizing Bellman errors computed along sampled trajectories. Here we focus on R2D2, a special instance of distributed value-based agent. Please refer to Kapturowski et al. (2019) for a complete review of all algorithmic details of value-based agents such as R2D2.

Across all baseline variants, the learner computes regression targets $Q_{\text{target}}(x, a) \approx Q^\pi(x, a)$ for the network to approximate $Q_\theta(x, a) \approx Q_{\text{target}}(x, a)$. The targets $Q_{\text{target}}(x, a)$ are calculated based on partial trajectories under μ which require off-policy corrections. We compare several correction variants: zero-order, first-order, Retrace (Munos et al., 2016; Rowland et al., 2020) and second-order. Please see algorithmic details in Appendix G.

The comparison results are in Figure 4 where we show the mean scores. We make several observations: **(1)** second-order correction leads to marginally better performance than first-order and retrace, and significantly better than zero-order. **(2)** In general, unbiased (or slightly biased) off-policy corrections do not yet perform as well as radically biased off-policy variants, such as *uncorrected-nstep* (Kapturowski et al., 2019; Rowland et al., 2020). **(3)** Zero-order performs the worst — though it is able to reach super human performance on most games as other variants but then the performance quickly plateaus. See Appendix H.7 for more results.

6. Discussion and conclusion

The idea of IS is the core of most off-policy evaluation techniques (Precup et al., 2000; Harutyunyan et al., 2016; Munos et al., 2016). We showed that Taylor expansions construct approximations to the full IS corrections and hence intimately relate to established off-policy evaluation techniques.

However, the connection between IS and policy optimization is less straightforward. Prior work focuses on applying off-policy corrections directly to policy gradient estimators (Jie and Abbeel, 2010; Espeholt et al., 2018) instead of the surrogate objectives which generate the gradients. Though standard policy optimization objectives (Schulman et al., 2015; 2017) involve IS weights, their link with IS is not made explicit. Closely related to our work is that of Tomczak et al. (2019), where they identified such optimization objectives as biased approximations to the full IS objective (Metelli et al., 2018). We characterized such approximations as the first-order special case of Taylor expansions and derived their natural generalizations.

In summary, we showed that Taylor expansions naturally connect trust-region policy search with off-policy evaluations. This new formulation unifies previous results, opens doors to new algorithms and bring significant gains to certain state-of-the-art deep RL agents.

Acknowledgements. Great thanks to Mark Rowland for insightful discussions during the development of ideas as well as extremely useful feedbacks on earlier versions of this paper. The authors also thank Diana Borsa, Jean-Bastien Grill, Florent Altché, Tadashi Kozuno, Zhongwen Xu, Steven Kapturowski, and Simon Schmitt for helpful discussions.

References

Abdolmaleki, A., Springenberg, J. T., Tassa, Y., Munos, R., Heess, N., and Riedmiller, M. (2018). Maximum a posteriori policy optimisation. In *International Conference on Learning Representations*.

Babaeizadeh, M., Frosio, I., Tyree, S., Clemons, J., and Kautz, J. (2017). Reinforcement learning through asynchronous advantage actor-critic on a gpu. *International Conference on Learning Representations*.

Barth-Maron, G., Hoffman, M. W., Budden, D., Dabney, W., Horgan, D., TB, D., Muldal, A., Heess, N., and Lillicrap, T. (2018). Distributional policy gradients. In *International Conference on Learning Representations*.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. (2013). The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279.

Berner, C., Brockman, G., Chan, B., Cheung, V., Debiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., et al. (2019). Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*.

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., Legg, S., and Kavukcuoglu, K. (2018). IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In *International Conference on Machine Learning*.

Gruslys, A., Dabney, W., Azar, M. G., Piot, B., Bellemare, M., and Munos, R. (2018). The reactor: A fast and sample-efficient actor-critic agent for reinforcement learning. In *International Conference on Learning Representations*.

Harutyunyan, A., Bellemare, M. G., Stepleton, T., and Munos, R. (2016). Q(λ) with Off-Policy Corrections. In *Algorithmic Learning Theory*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*.

Horgan, D., Quan, J., Budden, D., Barth-Maron, G., Hessel, M., van Hasselt, H., and Silver, D. (2018). Distributed prioritized experience replay. In *International Conference on Learning Representations*.

Jie, T. and Abbeel, P. (2010). On a connection between importance sampling and the likelihood ratio policy gradient. In *Neural Information Processing Systems*.

Kakade, S. and Langford, J. (2002). Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*.

Kapturowski, S., Ostrovski, G., Dabney, W., Quan, J., and Munos, R. (2019). Recurrent experience replay in distributed reinforcement learning. In *International Conference on Learning Representations*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Metelli, A. M., Papini, M., Faccio, F., and Restelli, M. (2018). Policy optimization via importance sampling. In *Neural Information Processing Systems*.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. In *NIPS Deep Learning Workshop*.

Munos, R., Stepleton, T., Harutyunyan, A., and Bellemare, M. (2016). Safe and efficient off-policy reinforcement learning. In *Neural Information Processing Systems*.

Nair, A., Srinivasan, P., Blackwell, S., Alcicek, C., Fearon, R., De Maria, A., Panneershelvam, V., Suleyman, M., Beattie, C., Petersen, S., et al. (2015). Massively parallel methods for deep reinforcement learning. *arXiv preprint arXiv:1507.04296*.

Pohlen, T., Piot, B., Hester, T., Azar, M. G., Horgan, D., Budden, D., Barth-Maron, G., Van Hasselt, H., Quan, J., Večerík, M., et al. (2018). Observe and look further: Achieving consistent performance on atari. *arXiv preprint arXiv:1805.11593*.

Precup, D., Sutton, R. S., and Singh, S. P. (2000). Eligibility traces for off-policy policy evaluation. In *International Conference on Machine Learning*.

Rowland, M., Dabney, W., and Munos, R. (2020). Adaptive trade-offs in off-policy learning. In *International Conference on Artificial Intelligence and Statistics*.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *International conference on machine learning*.

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

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

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503.

Song, H. F., Abdolmaleki, A., Springenberg, J. T., Clark, A., Soyer, H., Rae, J. W., Noury, S., Ahuja, A., Liu, S., Tirumala, D., Heess, N., Belov, D., Riedmiller, M., and Botvinick, M. M. (2020). V-MPO: on-policy maximum a posteriori policy optimization for discrete and continuous control. In *International Conference on Learning Representations*.

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

Tieleman, T. and Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31.

Tomczak, M. B., Kim, D., Vrancx, P., and Kim, K.-E. (2019). Policy optimization through approximated importance sampling. *arXiv preprint arXiv:1910.03857*.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. (2019). Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354.

Wang, Z., Bapst, V., Heess, N., Mnih, V., Munos, R., Kavukcuoglu, K., and de Freitas, N. (2017). Sample efficient actor-critic with experience replay. *International Conference on Learning Representations*.

Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., and Freitas, N. (2016). Dueling network architectures for deep reinforcement learning. In *International Conference on Machine Learning*.