Taylor expansion of discount factors

Yunhao Tang, Mark Rowland, Rémi Munos, Michal Valko

Taylor Expansions of Discount Factors

Yunhao Tang [1] **Mark Rowland** [2] **Rémi Munos** [3] **Michal Valko** [3]

Abstract

In practical reinforcement learning (RL), the discount factor used for estimating value functions often differs from that used for defining the evaluation objective. In this work, we study the effect that this discrepancy of discount factors has during learning, and discover a family of objectives that interpolate value functions of two distinct discount factors. Our analysis suggests new ways for estimating value functions and performing policy optimization updates, which demonstrate empirical performance gains. This framework also leads to new insights on commonly-used deep RL heuristic modifications to policy optimization algorithms.

1. Introduction

One of the most popular models for reinforcement learning (RL) is the Markov decision process (MDP) with exponential discounting over an infinite horizon (Sutton and Barto, 2018; Puterman, 2014), with discounted objectives of the following form

$$V_\gamma^\pi(x) = \mathbb{E}_\pi\left[\sum_{t=0}^\infty \gamma^t r_t \,\middle|\, x_0 = x\right].$$

Discounted models enjoy favorable theoretical properties, and are also the foundation of many practical RL algorithms that enjoy empirical success (e.g. see (Mnih et al., 2015; Schulman et al., 2015a; Lillicrap et al., 2015; Schulman et al., 2017)). However, in most applications of RL, the objective of interest is the expected *undiscounted cumulative return*,

$$\mathbb{E}_\pi\left[\sum_{t=0}^T r_t \,\middle|\, x_0 = x\right], \tag{1}$$

where $T < \infty$ is a (possibly random) evaluation horizon, which usually also denotes the end of the trajectory. For example, T could be the first time the MDP gets into a terminal state (e.g., a robot falls); when the MDP does not have a natural terminal state, T could be enforced as a deterministic horizon. This creates a technical gap between algorithmic developments and implementations: it is tempting to design algorithms that optimize $V_\gamma^\pi(x)$, however, further heuristics are often needed to get strong practical performance. This issue manifests itself with the policy gradient (PG) theorem (Sutton et al., 2000). Let π_θ be a parameterized policy. The policy gradient (PG) $\nabla_\theta V_\gamma^{\pi_\theta}(x)$ is computed as

$$\mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^\infty \gamma^t Q_\gamma^{\pi_\theta}(x_t, a_t)\nabla_\theta \log \pi_\theta(a_t|x_t) \,\middle|\, x_0 = x\right]. \tag{2}$$

However, the practical implementation of PG updates usually omits the discount factors (see for example the high-quality open source packages (Dhariwal et al., 2017; Achiam and OpenAI, 2018), leading to an approximate gradient of the form

$$\mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^T Q_\gamma^{\pi_\theta}(x_t, a_t)\nabla_\theta \log \pi_\theta(a_t|x_t) \,\middle|\, x_0 = x\right]. \tag{3}$$

Most prior work on PG algorithms rely on this heuristic update to work properly in deep RL applications. The intuitive argument for dropping the factor γ^t is that Eqn (2) optimizes $V_\gamma^{\pi_\theta}(x)$, which is very myopic compared to the objective in Eqn (1). Consequently, the exponential discount γ^t is too aggressive for weighting updates with large t. As a concrete example, in many MuJoCo control tasks (Brockman et al., 2016), the most commonly used discount factor is $\gamma = 0.99$. This leads to an effective horizon of $\frac{1}{1-\gamma} = 100$, which is much smaller than the evaluation horizon $T = 1000$. This technical gap between theory and practice has been alluded to previously (by e.g., O'Donoghue et al., 2016) and is explicitly discussed by Nota and Thomas (2019).

To bypass this gap, a straightforward solution would be to naïvely increase the discount factor $\gamma \geq 1 - \frac{1}{T}$ and apply the PG in Eqn (2). In the example above, this implies using $\gamma \geq 0.999$. Unfortunately, this rarely works well in practice, as we will also see in experiments. The failure might be due to the higher variance of the estimation (Schulman et al., 2015b) or the collapse of the action gaps (Lehnert et al., 2018; Laroche and van Seijen, 2018), which is aggravated when combined with function approximations.

[1]Columbia University, New York, USA [2]DeepMind, London, UK [3]DeepMind, Paris, France. Correspondence to: yt2541@columbia.edu <Yunhao>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

Nevertheless, as a theoretical framework, it is insightful to emulate the undiscounted objective in Eqn (1) using the (un)discounted objective $V^\pi_{\gamma'}(x)$ with $\gamma' \geq 1 - \frac{1}{T}$. To build intuitions about this approximation, note that when the time step is small $t \ll T$, the multiplicative factor $(\gamma')^t \approx 1$ and the cumulative rewards are almost undiscounted; even when $t = T$, we have $(\gamma')^t \geq (1 - \frac{1}{T})^T \approx \frac{1}{e} \gg 0$. Overall, this is a much more accurate approximation than $V^\pi_\gamma(x)$. This naturally prompts us to answer the following general question: *How do we evaluate and optimize $V^\pi_{\gamma'}(x)$ with estimates built for $V^\pi_\gamma(x)$ where $0 < \gamma < \gamma' \leq 1$?*

Main idea. We study the relation between $V^\pi_\gamma(x)$ and $V^\pi_{\gamma'}(x)$ via Taylor expansions. In Section 3, we identify a family of interpolating objectives between the more myopic objective $V^\pi_\gamma(x)$ and the true objective of interest $V^\pi_{\gamma'}(x)$. In Section 4, we start with insights on why the heuristic in Eqn (3) might be useful in practice. Then, we apply Taylor expansions directly to the heuristic updates, to arrive at a family of interpolating updates. In Section 5, we build on theoretical insights to derive improvements to established deep RL algorithms. We show their performance gains in Section 7.

2. Background

Consider the setup of a MDP. At any discrete time $t \geq 0$, the agent is in state $x_t \in \mathcal{X}$, takes an action $a_t \in \mathcal{A}$, receives an instant reward $r_t = r(x_t, a_t) \in [0, R_{\max}]$ and transitions to a next state $x_{t+1} \sim p(\cdot|x_t, a_t)$. For simplicity, we assume $r(x, a)$ to be deterministic. Let policy $\pi : \mathcal{X} \to \mathcal{P}(\mathcal{A})$ be a mapping from states to distributions over actions. Let $\gamma \in [0, 1)$ be a discount factor, define the Q-function $Q^\pi_\gamma(x, a) := \mathbb{E}_\pi \left[\sum_{t=0}^\infty \gamma^t r_t \mid x_0 = x, a_0 = a\right]$ and value function $V^\pi_\gamma(x) := \mathbb{E}_\pi \left[\sum_{t=0}^\infty \gamma^t r_t \mid x_0 = x\right]$. We also define the advantage function $A^\pi_\gamma(x, a) := Q^\pi_\gamma(x, a) - V^\pi_\gamma(x)$. Here, $\mathbb{E}_\pi[\cdot]$ denotes that the trajectories $(x_t, a_t, r_t)_{t=0}^\infty$ are generated under policy π. Throughout the paper, we use subscripts γ to emphasize that RL quantities implicitly depend on discount factors.

2.1. Linear programs for reinforcement learning

Henceforth, we assume all vectors to be column vectors. The value functions V^π_γ satisfy the Bellman equations $V^\pi_\gamma(x) = \mathbb{E}_\pi \left[r(x, a) + \gamma V^\pi_\gamma(x') \mid x_0 = x\right]$ (Bellman, 1957). Such equations can be encoded into a linear program (LP) (De Farias and Van Roy, 2003; Puterman, 2014). Let $V \in \mathbb{R}^\mathcal{X}$ be the primal variables, consider the following LP,

$$\max \ \delta_x^T V, \ \ V = r^\pi + \gamma P^\pi V, \tag{4}$$

where $r^\pi \in \mathbb{R}^\mathcal{X}$ is the state-dependent reward $r^\pi(x') := \sum_{a'} \pi(a'|x') r(x', a')$ and $P^\pi \in \mathbb{R}^{\mathcal{X} \times \mathcal{X}}$ is the transition matrix under π. Here, $\delta_x \in \mathbb{R}^\mathcal{X}$ encodes the one-hot distribution (Dirac) at x. Similar results hold for considering the LP objective $v^T V$ with a general distribution $v \in \mathcal{P}(\mathcal{X})$. It then follows that the optimal solution to the above LP is $V^* = V^\pi_\gamma$. Now, consider the dual LP to Eqn (4), let $d \in \mathbb{R}^\mathcal{X}$ be the dual variables,

$$\min \ (1 - \gamma)^{-1} (r^\pi)^T d, \ \ d = (1 - \gamma)\delta_x + \gamma (P^\pi)^T d. \tag{5}$$

The optimal solution to the dual program has a natural probabilistic interpretation. It is the discounted visitation distribution $d^\pi_{x,\gamma}$ under policy π with starting state x as $d^\pi_{x,\gamma}(x') := (1 - \gamma) \sum_{t \geq 0} \gamma^t P_\pi(x_t = x' | x_0 = x)$ where $P_\pi(x_t = x' | x_0 = x)$ is a probability measure induced by the policy π and the MDP transition kernel. By strong duality, the value function can be equivalently written as

$$V^\pi_\gamma(x) = \frac{1}{1 - \gamma} \mathbb{E}_{x' \sim d^\pi_{x,\gamma}, a' \sim \pi(\cdot|x')} \left[r(x', a')\right]. \tag{6}$$

3. Taylor Expansions of Value Functions

Below, we show how to estimate $V^\pi_{\gamma'}(x)$ with approximations constructed from value functions $V^\pi_\gamma(x)$ for $\gamma < \gamma'$. Unless otherwise stated, we always assume $\gamma' < 1$ for a more convenient mathematical treatment of the problem.

3.1. Taylor expansions of discount factors

We start with some notations: we abuse the notation of value functions $V^\pi_\gamma \in \mathbb{R}^\mathcal{X}$ to both refer to the scalar function as well as a vector. The Bellman equation for the value-function is expressed in the matrix form (Puterman, 2014)

$$V^\pi_{\gamma'} = r^\pi + \gamma' P^\pi V^\pi_{\gamma'}. \tag{7}$$

Inverting the equation,

$$V^\pi_{\gamma'} = (I - \gamma' P^\pi)^{-1} r^\pi. \tag{8}$$

Now, we present the main result of Taylor expansions.

Proposition 3.1. The following holds for all $K \geq 0$,

$$V^\pi_{\gamma'} = \sum_{k=0}^K \left((\gamma' - \gamma)(I - \gamma P^\pi)^{-1} P^\pi\right)^k V^\pi_\gamma + \underbrace{\left((\gamma' - \gamma)(I - \gamma P^\pi)^{-1} P^\pi\right)^{K+1} V^\pi_{\gamma'}}_{\text{residual}}. \tag{9}$$

When $\gamma < \gamma' < 1$, the residual norm converges to 0, which implies

$$V^\pi_{\gamma'} = \sum_{k=0}^\infty \left((\gamma' - \gamma)(I - \gamma P^\pi)^{-1} P^\pi\right)^k V^\pi_\gamma. \tag{10}$$

We provide a proof sketch here: Note that $\gamma' P^\pi = (\gamma' - \gamma)P^\pi + \gamma P^\pi$ and apply the Woodbury matrix identity to obtain $(I - \gamma' P^\pi)^{-1} = (I - \gamma P^\pi)^{-1} + (\gamma' - \gamma)(I - \gamma P^\pi)^{-1} P^\pi (I - \gamma' P^\pi)^{-1}$. We can then recursively expand Eqn (8) K times to arrive at Eqn (9). In particular, by expanding the equation once, we see that $(I - \gamma' P^\pi)^{-1}$ is equivalent to the following,

$$(I - \gamma P^\pi)^{-1} + (\gamma' - \gamma)(I - \gamma P^\pi)^{-1} P^\pi (I - \gamma P^\pi)^{-1} + (\gamma' - \gamma)^2 \left((I - \gamma P^\pi)^{-1} P^\pi\right)^2 \underbrace{(I - \gamma' P^\pi)^{-1}}_{\text{can be expanded further}},$$

where the last term can be expanded further by plugging in the Woodbury matrix identity. See the complete proof in Appendix A.

Extensions to $\gamma' = 1$. The above result can extend to the case $\gamma' = 1$. We make two assumptions: **A.1** The Markov chain induced by π is absorbing and T is the absorption time; **A.2** $r^\pi(x) = 0$ for absorbing states x. Under these assumptions, we can interpret such absorbing states as the terminal states. As a result, $V^\pi_{\gamma'=1}(x) = \mathbb{E}_\pi \left[\sum_{t=0}^T r_t \,\middle|\, x_0 = x\right]$ is well-defined and Proposition 3.1 still holds; see Appendix A for the complete proof.

In practice, it is infeasible to sum up all infinite number of terms in the Taylor expansion. It is then of interest to consider the K^{th}-order expansion of $V^\pi_{\gamma'}$, which truncates the infinite series. Specifically, we define the K^{th}-order expansion as

$$V^\pi_{K,\gamma,\gamma'} \coloneqq \sum_{k=0}^{K} ((\gamma' - \gamma)(I - \gamma P^\pi)^{-1} P^\pi)^k V^\pi_\gamma \,. \quad (11)$$

As K increases, the K^{th} order expansion becomes increasingly close to the infinite series, which evaluates to $V^\pi_{\gamma'}(x)$. This is formalized next.

Proposition 3.2. The following bound holds for all $K \geq 0$,

$$\left|V^\pi_{\gamma'}(x) - V^\pi_{K,\gamma,\gamma'}(x)\right| \leq \left(\frac{\gamma' - \gamma}{1 - \gamma}\right)^{K+1} \frac{R_{\max}}{1 - \gamma'}. \quad (12)$$

3.2. Sample-based approximations of Taylor expansions

We now describe how to estimate $V^\pi_{K,\gamma,\gamma'}(x)$ via samples. First, we build some intuition on the behavior of expansions at different orders K by considering a few special cases.

Zeroth-order expansion. By setting $K = 0$, we see that

$$V^\pi_{0,\gamma,\gamma'} = V^\pi_\gamma. \quad (13)$$

The zeroth order expansion approximates the value function $V^\pi_{\gamma'}(x)$ of the discount factor γ' with that $V^\pi_\gamma(x)$ of a lower discount factor $\gamma < \gamma'$. This is a very straightforward approximation to use in that no sampling at all is required, but it may not be accurate.

First-order expansion. When $K = 1$, we consider the increments of the expansions,

$$V^\pi_{1,\gamma,\gamma'} - V^\pi_{0,\gamma,\gamma'} = (\gamma' - \gamma)(I - \gamma P^\pi)^{-1} P^\pi V^\pi_\gamma. \quad (14)$$

To understand the first order expansion, recall that in the definition of value function $V^\pi_\gamma = (I - \gamma P^\pi)^{-1} r^\pi$, immediate rewards r^π are *accumulated* via the matrix $(I - \gamma P^\pi)^{-1}$. In general, for any $X, Y \in \mathbb{R}^{\mathcal{X}}$, we can interpret $X = (I - \gamma P^\pi))^{-1} Y$ as accumulating Y as rewards to compute X as value functions. By analogy, we can interpret the RHS of Eqn (14) as the value function assuming $(\gamma' - \gamma)P^\pi V^\pi_\gamma$ as immediate rewards. In other words, the first order expansion bootstraps the zeroth order expansion V^π_γ to form a more accurate approximation. Combined with the zeroth order expansion, we can also conveniently write the difference of first- and zeroth-order expansions as an expectation $V^\pi_{1,\gamma,\gamma'}(x) - V_{0,\gamma,\gamma'}(x) = (\gamma' - \gamma)\mathbb{E}_\pi \left[\sum_{t=1}^\infty \gamma^{t-1} V^\pi_\gamma(x_t) \,\middle|\, x_0 = x\right]$. Let $\tau \sim \text{Geometric}(1 - \gamma)$ be a random time such that $P(\tau = t) = (1 - \gamma)\gamma^t, \forall t \in \mathbb{Z}_{\geq 1}$. The difference can also be expressed via this random time

$$V^\pi_{1,\gamma,\gamma'}(x) - V_{0,\gamma,\gamma'}(x) = \frac{\gamma' - \gamma}{1 - \gamma} \mathbb{E}_{\pi,\tau} \left[V^\pi_\gamma(x_\tau)\right].$$

Note that from this expression, we obtain a simple unbiased estimate for $V^\pi_{1,\gamma,\gamma'}(x) - V_{0,\gamma,\gamma'}(x)$, using a sampled trajectory and a random time step τ.

General K^{th}-order expansion. We now present results for general K. Consider the incremental term,

$$V^\pi_{K,\gamma,\gamma'} - V^\pi_{K-1,\gamma,\gamma'} = (\gamma' - \gamma)^K \left((I - \gamma P^\pi)^{-1} P^\pi\right)^K V^\pi_\gamma. \quad (15)$$

Note that the aggregate matrix $\left((I - \gamma P^\pi)^{-1} P^\pi\right)^K$ suggests a recursive procedure to bootstrap from lower order expansions to construct higher order expansions. To see why, we can rewrite the right-hand side of Eqn (15) as

$$(\gamma' - \gamma)(I - \gamma P^\pi)^{-1} P^\pi \left(V^\pi_{K-1,\gamma,\gamma'} - V^\pi_{K-2,\gamma,\gamma'}\right).$$

Indeed, we can interpret the difference $V^\pi_{K,\gamma,\gamma'} - V^\pi_{K-1,\gamma,\gamma'}$ as the value function under the immediate reward $(\gamma' - \gamma)P^\pi \left(V^\pi_{K-1,\gamma,\gamma'} - V^\pi_{K-2,\gamma,\gamma'}\right)$. This generalizes the bootstrap procedure of the first order expansion as a special case where we naturally assume $V^\pi_{-1,\gamma,\gamma'} = 0$. Given K

i.i.d. random times $\tau_i \sim \text{Geometric}(1-\gamma)$, we can write $V^\pi_{K,\gamma,\gamma'}(x) - V^\pi_{K-1,\gamma,\gamma'}(x)$ as the expectation

$$\left(\frac{\gamma'-\gamma}{1-\gamma}\right)^K \mathbb{E}_{\tau_i, 1\le i\le K}\left[V^\pi_\gamma\left(x_{\tau_1+\cdots+\tau_K}\right)\right].$$

Based on the above expression, Algorithm 1 provides a subroutine that generates unbiased estimates of $V^\pi_{K,\gamma,\gamma'}(x)$ by sub-sampling an infinite trajectory $(x_t, a_t, r_t)_{t=0}^\infty$ with the random times.

Practical implementations. While the above and Algorithm 1 show how to compute one-sample estimates, in practice, we might want to average multiple samples along a single trajectory for variance reduction. See Appendix F for further details on the practical estimates.

Algorithm 1 Estimating the K^{th} order expansion

Require: A trajectory $(x_t, a_t, r_t)_{t=0}^\infty \sim \pi$ and discount factors $\gamma < \gamma' < 1$

1. Compute an unbiased estimate $\widehat{V}^\pi_\gamma(x_t)$ for states along the trajectory, e.g., $\widehat{V}^\pi_\gamma(x_t) = \sum_{t'\ge t} \gamma^{t'-t} r_{t'}$.
2. Sample K random time $\{\tau_i\}_{1\le i\le K}$, all i.i.d. geometrically distributed $\tau_i \sim \text{Geometric}(1-\gamma)$.
3. Return the unbiased estimate $\sum_{k=0}^K \left(\frac{\gamma'-\gamma}{1-\gamma}\right)^k \widehat{V}^\pi_\gamma(x_{t_k})$ where $t_k = \sum_{i=1}^k \tau_i$.

Interpretation of expansions in the dual space. Recall that $V^\pi_{\gamma'} = (I-\gamma' P^\pi)^{-1} r^\pi = I(I-\gamma' P^\pi)^{-1} r^\pi$ where the identity matrix $I = [\delta_0, \delta_1, ...\delta_{\mathcal{X}}]$ concatenates Dirac delta vectors $\delta_x, \forall x \in \mathcal{X}$. Since r^π is a constant vector, Taylor expansions essentially construct approximations to the matrix $(I-\gamma' P^\pi)^{-1}$. By grouping the matrix with the reward vector (or the density matrix), we arrive at the primal expansion (or the dual expansion),

$$I \underbrace{(I-\gamma' P^\pi)^{-1} r^\pi}_{\text{primal expansions of } V^\pi_{\gamma'}(x)} = \underbrace{I(I-\gamma' P^\pi)^{-1}}_{\text{dual expansions of } d^\pi_{x,\gamma'}} r^\pi$$

The derivations above focus on the primal expansion view. We show a parallel theory of dual expansion in Appendix B. The equivalence of primal-dual view of Taylor expansions suggests connections with seemingly disparate lines of prior work: Janner et al. (2020) propose a density model for visitation distribution of different γ in the context of model-based RL. They show that predictions of large discount factors could be bootstrapped from predictions of small discount factors. This corresponds exactly to the dual space expansions, which is equivalent to the primal space expansions.

Extensions to Q-functions. In Appendix C, we show that it is possible to build approximations to $Q^\pi_{\gamma'}$ using Q^π_γ as building blocks. The theoretical guarantees and estimation procedures are similar to the case of value functions.

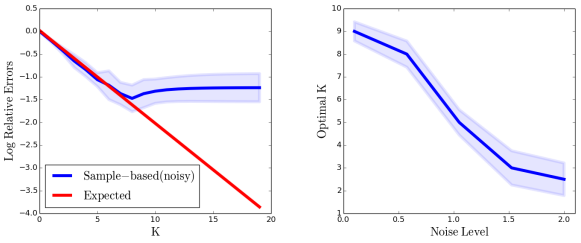
3.3. Approximation errors with finite samples

Proposition 3.2 shows that the *expected* approximation error decays as $\left|V^\pi_{K,\gamma,\gamma'}(x) - V^\pi_{\gamma'}(x)\right| = O\left(\left(\frac{\gamma'-\gamma}{1-\gamma}\right)^{K+1}\right)$ for $\gamma < \gamma' < 1$. This motivates using a high value of K when constructing the approximation. However, in practice, all constituent terms in the K^{th} order expansion are random estimates, each with a non-zero variance. This might lead the variance of the overall estimate to increase as K increases. As a result, K mediates a trade-off between bias (expected approximation error) and variance. We formalize such intuitions in Appendix E, where we theoretically analyze the trade-off using the phased TD-learning framework (Kearns and Singh, 2000).

A numerical example. To get direct intuition about the effect of K, we focus on a tabular MDP example. The MDP has $|\mathcal{X}| = 10$ states and $|\mathcal{A}| = 2$ actions. All entries of the transition table $p(y|x,a)$ are generated from a Dirichlet distribution with parameters $(\alpha, \ldots, \alpha)$ with $\alpha = 0.01$. The policy $\pi(a|x)$ is uniformly random. We take $\gamma = 0.2$ and $\gamma' = 0.8$. The agent generates $N = 10$ trajectories $(x_t, a_t, r_t)_{t=0}^T$ with a very large horizon T with a fixed starting state x_0. We assume access to base estimates $\widehat{V}^\pi_\gamma(x_t)$ and the Taylor expansion estimates $\widehat{V}^\pi_{K,\gamma,\gamma'}(x_0)$ are computed based on Algorithm 1. We estimate the relative error as $\widehat{E}_K(x_0) = \left|V^\pi_{\gamma'}(x_0) - \widehat{V}^\pi_{K,\gamma,\gamma'}(x_0)\right|$. For further experiment details, see Appendix F.

In Figure 1(a), we show how errors vary as a function of K. We study two settings: **(1)** Expected estimates (red), where $\widehat{V}^\pi_{K,\gamma,\gamma'}(x_0)$ is computed analytically through access to transition tables. In this case, similar to how the theory suggests, the error decays exponentially; **(2)** Sample-based estimates (blue) with base estimates $\widehat{V}^\pi_\gamma(x_t) = \sum_{s=0}^\infty \gamma^s r_{t+s}$. The errors decay initially with K but later start to increase a bit as K gets large. The optimal K in the middle achieves the best bias-variance trade-off. Note that in this particular example, the estimates do not pay a very big price in variance for large K. We speculate this is because increments to the estimates are proportional to $\left(\frac{\gamma'-\gamma}{1-\gamma}\right)^{K+1}$, which scales down additional variance terms quickly as K increases.

In Figure 1(b), we study how the optimal expansion order K^* depends on the noise level of base estimates. To emulate the noise, we assume access to base estimates $\widehat{V}^\pi_\gamma(x_t) = V^\pi_\gamma(x_t) + \mathcal{N}(0, \sigma^2)$ for some noise level σ. The optimal order K^* is computed as $K^* = \arg\min_k \widehat{E}_k(x_0)$. In general, we observe that when σ increases, K^* decreases. Intuitively, this implies that as the base estimates $\widehat{V}^\pi_\gamma(x)$ become noisy, we should prefer smaller value of K to control the variance. This result bears some insights for practical applications such as downstream policy optimization, where



(a) Trade-off of K (b) Optimal K

Figure 1. Comparison of Taylor expansions with different orders. The x-axis shows the order K, the y-axis shows the log relative errors of the approximations. The blue curve shows the exact computations while the red curve shows the sample based estimations. See Appendix F for more details.

we need to select an optimal K for the tasks at hand.

4. Taylor Expansions of Gradient Updates

In Section 3, we discussed how to construct approximations to $V^{\pi}_{\gamma'}(x)$. For the purpose of policy optimization, it is of direct interest to study approximations to $\nabla_\theta V^{\pi_\theta}_{\gamma'}(x)$. As stated in Section 1, a major premise of our work is that in many practical contexts, estimating discounted values under $\gamma' \approx 1$ is difficult. As a result, directly evaluating the full gradient $\nabla_\theta V^{\pi_\theta}_{\gamma'}(x)$ is challenging, because it requires estimating Q-functions $Q^{\pi_\theta}_{\gamma'}(x,a)$. Below, we start by showing how the decomposition of $\nabla_\theta V^{\pi_\theta}_{\gamma'}(x)$ motivates a particular form of gradient update, which is generally considered a deep RL heuristic. Then we construct approximations to this update based on Taylor expansions.

4.1. $V^{\pi}_{\gamma'}$ as a weighted mixture of V^{π}_{γ}

We can explicitly rewrite $V^{\pi}_{\gamma'}(x)$ as a weighted mixture of value functions $V^{\pi}_{\gamma}(x'), x' \in \mathcal{X}$. This result was alluded to in (Romoff et al., 2019) and formally shown below.

Lemma 4.1. Assume $\gamma < \gamma' < 1$. We can write $V^{\pi}_{\gamma'}(x) = (\rho^{\pi}_{x,\gamma,\gamma'})^T V^{\pi}_{\gamma}$, where the weight vector $\rho^{\pi}_{x,\gamma,\gamma'} \in \mathbb{R}^{\mathcal{X}}$ is

$$\left(I - \gamma (P^\pi)^T\right)\left(I - \gamma' (P^\pi)^T\right)^{-1} \delta_x.$$

Also we can rewrite $V^{\pi}_{\gamma'}(x)$, using an expectation, as:

$$V^{\pi}_{\gamma}(x) + \mathbb{E}_\pi\left[\sum_{t=1}^{\infty} (\gamma' - \gamma)(\gamma')^{t-1} V^{\pi}_{\gamma}(x_t) \,\middle|\, x_0 = x\right]. \tag{16}$$

When $\gamma' = 1$, $\rho^{\pi}_{x,\gamma,\gamma'}$ might be undefined. However, Eqn (16) still holds if assumptions **A.1** and **A.2** are satisfied.

4.2. Decomposing the full gradient $\nabla_\theta V^{\pi_\theta}_{\gamma'}(x)$

Lemma 4.1 highlights that $V^{\pi}_{\gamma'}(x)$ depends on π in two aspects: **(1)** the value functions $V^{\pi}_{\gamma}(x'), x' \in \mathcal{X}$; **(2)** the state-dependent distribution $\rho^{\pi}_{x,\gamma,\gamma'}(x')$. Let π_θ be a parameterized policy. For conceptual clarity, we can write $V^{\pi_\theta}_{\gamma'}(x) = F(V^{\pi_\theta}_{\gamma}, \rho^{\pi_\theta}_{x,\gamma,\gamma'})$ with a function $F : \mathbb{R}^{\mathcal{X}} \times \mathbb{R}^{\mathcal{X}} \to \mathbb{R}$. Though this function is essentially the inner product, i.e., $F(V,\rho) = V^T\rho$, notationally, it helps stress that $V^{\pi_\theta}_{\gamma'}(x)$ depends on θ through two vector arguments. Now, we can decompose $\nabla_\theta V^{\pi_\theta}_{\gamma'}(x)$.

Lemma 4.2. The full gradient $\nabla_\theta V^{\pi_\theta}_{\gamma'}(x)$ can be decomposed into the sum of two partial gradients as follows,

$$\begin{aligned}&\left(\partial_V F(V,\rho)\right)^T \nabla_\theta V^{\pi_\theta}_{\gamma} + \left(\partial_\rho F(V,\rho)\right)^T \nabla_\theta \rho^{\pi_\theta}_{x,\gamma,\gamma'} \\ &= \underbrace{\mathbb{E}\left[\nabla_\theta V^{\pi_\theta}_{\gamma}(x')\right]}_{\text{first partial gradient}} + \underbrace{\mathbb{E}\left[V^{\pi_\theta}_{\gamma}(x') \nabla_\theta \log \rho^{\pi_\theta}_{x,\gamma,\gamma'}(x')\right]}_{\text{second partial gradient}},\end{aligned}$$

where the above partial gradients are both evaluated at $V = V^{\pi_\theta}_{\gamma}, \rho = \rho^{\pi_\theta}_{x,\gamma,\gamma'}$ and both expectations are with respect to $x' \sim \rho^{\pi_\theta}_{x,\gamma,\gamma'}$.

We argue that the second partial gradient introduces most challenges in practical optimization. Intuitively, this is because its unbiased estimator is equivalent to a REINFORCE gradient estimator which requires estimating discounted values that accumulate $V^{\pi}_{\gamma}(x')$ as 'reward' under discount factor γ'. By the premise of our work, this estimation would be difficult. We will detail the discussions in Appendix D.

The following result characterizes the first partial gradient.

Proposition 4.3. For any $\gamma < \gamma' < 1$, the first partial gradient $(\partial_V F(V^{\pi_\theta}_{\gamma}, \rho^{\pi_\theta}_{x,\gamma,\gamma'}))^T \nabla_\theta V^{\pi_\theta}_{\gamma}$ can be expressed as

$$\mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} (\gamma')^t Q^{\pi_\theta}_{\gamma}(x_t, a_t) \nabla_\theta \log \pi_\theta(a_t|x_t) \,\middle|\, x_0 = x\right]. \tag{17}$$

When $\gamma' = 1$, under assumptions **A.1** and **A.2**, the first partial gradient exists and is expressed as

$$\mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T} Q^{\pi_\theta}_{\gamma}(x_t, a_t) \nabla_\theta \log \pi_\theta(a_t|x_t) \,\middle|\, x_0 = x\right]. \tag{18}$$

Connections to common deep RL heuristic. Many high-quality deep RL algorithms (see, e.g. Dhariwal et al., 2017; Achiam and OpenAI, 2018) implement parameter updates which are very similar to Eqn (18). As such, Proposition 4.3 provides some insights on why implementing such a heuristic might be useful in practice: though in general Eqn (18) is not a gradient (Nota and Thomas, 2019), it is a partial gradient of $V^{\pi_\theta}_{\gamma'=1}(x)$, which is usually the objective of interest at evaluation time. Compared with the formula of vanilla

PG in Eqn (2), Eqn (18) offsets the *over-discounting* by via a uniform average over states.

However, it is worth noting that in deep RL practice, the definition of the evaluation horizon T might slightly differ from that specified in **A.1**. In such cases, Proposition 4.3 does not hold. By **A.1**, T is the absorption time that defines when the MDP enters a terminal absorbing state. In many applications, however, for MDPs without a natural terminatal state, T is usually enforced by an external time constraint which does not depend on states. In other words, an environment can terminate even when it does not enter any terminal state (see, e.g., Brockman et al., 2016 for such examples). To bypass this subtle technical gap, one idea is to incorporate time steps as part of the state $\widetilde{x} \leftarrow [x, t]$. This technique was hinted at in early work such as (Schulman et al., 2015b) and empirically studied in (Pardo et al., 2018). In this case, the random absorbing time T depends fully on the augmented states, and Proposition 4.3 holds.

4.3. Taylor expansions of partial gradients

We now consider approximations to the first partial gradients

$$\left(\partial_V F(V_\gamma^{\pi_\theta}, \rho_{x,\gamma,\gamma'}^{\pi_\theta})\right)^T \nabla_\theta V_\gamma^{\pi_\theta} = (\rho_{x,\gamma,\gamma'}^{\pi_\theta})^T \nabla_\theta V_\gamma^{\pi_\theta}.$$

Since $\nabla_\theta V_\gamma^{\pi_\theta}$ does not depend on γ', the approximation is effectively with respect to the weight vector $\rho_{x,\gamma,\gamma'}^{\pi_\theta}$. Below, we show results for the K^{th} order approximation.

Proposition 4.4. Assume $\gamma < \gamma' < 1$. For any $x \in \mathcal{X}$, define the K^{th} Taylor expansion to $\rho_{x,\gamma,\gamma'}^{\pi}$ as

$$\rho_{x,K,\gamma,\gamma'}^{\pi} = \sum_{k=0}^{K} \left((\gamma' - \gamma)\left(I - \gamma(P^\pi)^T\right)^{-1}(P^\pi)^T\right)^k \delta_x.$$

It can be shown that $V_{K,\gamma,\gamma'}^{\pi}(x) = (\rho_{x,K,\gamma,\gamma'}^{\pi})^T V_\gamma^\pi$ and $\left\|\rho_{x,K,\gamma,\gamma'}^{\pi} - \rho_{K,\gamma,\gamma'}^{\pi}\right\|_\infty = O\left(\left(\frac{\gamma'-\gamma}{1-\gamma}\right)^{K+1}\right)$.

We build some intuitions about the approximations. Note that in general we can write the partial gradient as a weighted mixture of *local gradients* $Q_t \nabla_\theta \log \pi_\theta(a_t|x_t)$ where $Q_t := Q_\gamma^{\pi_\theta}(x_t, a_t)$,

$$\mathbb{E}_\pi\left[\sum_{t=0}^{\infty} w_{K,\gamma,\gamma'}(t) Q_t \nabla_\theta \log \pi_\theta(a_t|x_t) \,\middle|\, x_0 = x\right], \quad (19)$$

for some weight function $w_{K,\gamma,\gamma'}(t) \in \mathbb{R}$. When $K \to \infty$, $\lim w_{K,\gamma,\gamma'}(t) = (\gamma')^t$ and we recover the original first partial gradient defined in Eqn (17); when $K = 0$, $w_{K,\gamma,\gamma'}(t) = \gamma^t$ recovers the vanilla PG in Eqn (2). For other values of K, we show the analytic weights $w_{K,\gamma,\gamma'}(t)$ in Appendix D. Similar to how $V_{K,\gamma,\gamma'}^{\pi}$ interpolates V_γ^π and $V_{\gamma'}^\pi$, here the K^{th} order expansion to the partial gradients interpolate the full partial gradients and vanilla PG. In practice, we might expect an intermediate value of K achieve the best bias and variance trade-off of the update.

5. Policy optimization with Taylor expansions

Based on theoretical insights of previous sections, we propose two algorithmic changes to baseline algorithms. Based on Section 3, we propose Taylor expansion advantage estimation; based on Section 4, we propose Taylor expansion update weighting. It is important to note that other algorithmic changes are possible, which we leave to future work.

5.1. Baseline near on-policy algorithm

We briefly introduce backgrounds for near on-policy policy optimization algorithms (Schulman et al., 2015a; Mnih et al., 2016; Schulman et al., 2017; Espeholt et al., 2018). We assume that the data are collected under a behavior policy $(x_t, a_t, r_t)_{t=0}^{\infty} \sim \mu$, which is close to the target policy π_θ. The on-policyness is ensured by constraining $D(\pi_\theta, \mu) \le \varepsilon$ for some divergence D and threshold $\varepsilon > 0$. Usually, ε is chosen to be small such that little off-policy corrections are needed for estimating value functions. With data $(x_t, a_t, r_t)_{t=0}^{\infty}$, the algorithms estimate Q-functions $\widehat{Q}_\gamma^{\pi_\theta} \approx Q_\gamma^{\pi_\theta}$. Then the estimates $\widehat{Q}_\gamma^{\pi_\theta}(x, a)$ are used as plug-in alternatives to the Q-functions in the definition of gradient updates such as Eqn (2) for sample-based updates.

5.2. Taylor expansion Q-function estimation

In Section 3, we discussed how to construct approximations to $Q_{\gamma'}^{\pi_\theta}$ using $Q_\gamma^{\pi_\theta}$ as building blocks. As the first first algorithmic change, we propose to construct the K^{th} order expansion $Q_{K,\gamma,\gamma'}^{\pi_\theta}$ as a plug-in alternative to $Q_\gamma^{\pi_\theta}$ when combined with downstream optimization. Since $Q_{K,\gamma,\gamma'}^{\pi_\theta} \approx Q_{\gamma'}^{\pi_\theta}$, we expect the optimization subroutine to account for an objective of a longer effective horizon.

In many baseline algorithms, we have access to a value function critic $V_\phi(x)$ and a subroutine which produces Q-function estimates $\widehat{Q}_\gamma^{\pi_\theta}(x, a)$ (e.g., $\widehat{Q}_\gamma^{\pi_\theta}(x_t, a_t) = \sum_{s=0}^{\infty} \gamma^s r_{t+s}$). We then construct the K^{th} order expansion $\widehat{Q}_{K,\gamma,\gamma'}^{\pi_\theta}(x, a)$ using $\widehat{Q}_\gamma^{\pi_\theta}$. This procedure is similar to Algorithm 1 and we show the full algorithm in Appendix C. See also Appendix F for further experimental details.

5.3. Taylor expansion update weighting

In Section 4, we discussed Taylor expansions approximation $\rho_{x,K,\gamma,\gamma'}^{\pi_\theta}$ to the weight vector $\rho_{x,\gamma,\gamma'}^{\pi_\theta}$. As the second algorithmic change to the baseline algorithm, we update parameters in the direction of K^{th} order approximations to the partial gradient $\theta \leftarrow \theta + \alpha \left(\rho_{x,K,\gamma,\gamma'}^{\pi_\theta}\right)^T \nabla_\theta V_\gamma^{\pi_\theta}$. Eqn (19) shows that the update effectively translates into adjusting the weight $w_t = w_{K,\gamma,\gamma'}(t)$. When combined with other

Algorithm 2 Taylor expansion Q-function estimation

Require: policy π_θ with parameter θ and α
while not converged **do**
1. Collect partial trajectories $(x_t, a_t, r_t)_{t=1}^T \sim \mu$.
2. Estimate Q-functions $\widehat{Q}_\gamma^{\pi_\theta}(x_t, a_t)$.
3. Construct K^{th} order Taylor expansion estimator $\widehat{Q}_{K,\gamma,\gamma'}^{\pi_\theta}(x_t, a_t)$ using $\widehat{Q}^{\pi_\theta}(x_t, a_t)$.
4. Update the parameter via gradient ascent $\theta \leftarrow \theta + \alpha \sum_{t=1}^T \widehat{Q}_{K,\gamma,\gamma}^{\pi_\theta}(x_t) \nabla_\theta \log \pi_\theta(a_t|x_t)$.

end while

components of the algorithm, the pseudocode is shown in Algorithm 3. Under this framework, the common deep RL heuristic could be recovered by setting $w_t = 1$.

Algorithm 3 Taylor expansion update weighting

Require: policy π_θ with parameter θ and α
while not converged **do**
1. Collect partial trajectories $(x_t, a_t, r_t)_{t=1}^T \sim \mu$.
2. Estimate Q-functions $\widehat{Q}_t = \widehat{Q}_\gamma^{\pi_\theta}(x_t, a_t)$.
3. Compute weights for each state $w_t = w_{x_0,K,\gamma,\gamma'}(t)$, and average $g_\theta = \sum_{t=1}^T w_t \widehat{Q}_t \nabla_\theta \log \pi_\theta(a_t|x_t)$.
4. Update parameters $\theta \leftarrow \theta + \alpha g_\theta$.

end while

6. Related work

Discount factors in RL. Discount factors impact RL agents in various aspects. A number of work suggest that RL problems with large discount factors are generally more difficult to solve (Jiang et al., 2016), potentially due to increased complexities of the optimal value functions or collapses of the action gaps (Lehnert et al., 2018; Laroche and van Seijen, 2018). However, optimal policies defined with small discounts can be very sub-optimal for RL objectives with a large discount factor. To entail numerical stability of using large discounts, prior work has suggested non-linear transformation of the Bellman targets for Q-learning (Pohlen et al., 2018; van Hasselt et al., 2019; Kapturowski et al., 2018; Van Seijen et al., 2019). However, when data is scarce, small discount factors might prove useful due to its implicit regularization effect (Amit et al., 2020).

As such, there is a trade-off mediated by choosing different values of discount factors. Similar trade-off effects are most well-known in the context of TD(λ), where $\lambda \in [0, 1]$ trades-off the bias and variance of the TD updates (Sutton and Barto, 2018; Kearns and Singh, 2000).

Adapting discount factors & multiple discount factors. In general, when selecting a single optimal discount factor for training is difficult, it might be desirable to adjust the discount during training. This could be achieved by human-designed (Prokhorov and Wunsch, 1997; François-Lavet et al., 2015) or blackbox adaptation (Xu et al., 2018). Alternatively, it might also be beneficial to learn with multiple discount factors at the same time, which could improve TD-learning (Sutton, 1995) or representation learning (Fedus et al., 2019). Complementary to all such work, we study the connections between value functions defined with different discounts.

Taylor expansions for RL. Recently in (Tang et al., 2020), Taylor expansions were applied to study the relationship between V_γ^π and V_γ^μ, i.e., value functions under the same discount factor but different policies $\pi \neq \mu$. This is useful in the context of off-policy learning. Our work is orthogonal and could be potentially combined with this approach.

7. Experiments

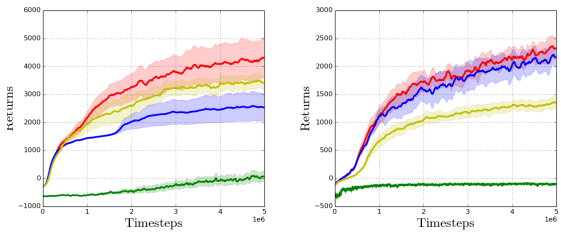
In this section, we evaluate the empirical performance of new algorithmic changes to the baseline algorithms. We focus on robotics control experiments with continuous state and action space. The tasks are available in OpenAI gym (Brockman et al., 2016), with backends such as MuJoCo (Todorov et al., 2012) and bullet physics (Coumans, 2015). We label the tasks as gym (G) and bullet (B) respectively. We always compare the undiscounted cumulative rewards evaluated under a default evaluation horizon $T = 1000$.

Hyper-parameters. Throughout the experiments, we use the same hyper-parameters across all algorithms. The learning rate is tuned for the baseline PPO, and fixed across all algorithms. See Appendix F for further details.

7.1. Taylor expansion Q-function estimation

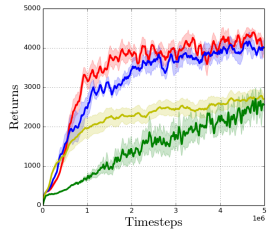
We use $\widehat{Q}_{K,\gamma,\gamma'}^{\pi_\theta}(x, a)$ with $K = 1$ as the Q-function estimator plug-in for the gradient update. When combining with PPO (Schulman et al., 2017), the resulting algorithm is named PPO(K). We compare with the baseline PPO and TRPO (Schulman et al., 2015a). In practice, we consider a mixture of advantage estimator $\widehat{Q}^{\pi_\theta}(x, a) = (1 - \eta)\widehat{Q}_\gamma^{\pi_\theta}(x, a) + \eta \widehat{Q}_{K,\gamma,\gamma'}^{\pi_\theta}(x, a)$ with $\eta \in [0, 1]$ a constant that interpolates between the PPO (i.e., $\eta = 0$) and PPO(K). Note that though η should be selected such that it balances the numerical scales of the two extremes, as a result, we usually find η to work well when it is small in absolute scale ($\eta = 0.01$ works the best).

Results. In Figure 2, we compare a few baselines: **(1)** PPO with $\gamma = 0.99$ (default); **(2)** PPO with high discount factor $\gamma = 1 - \frac{1}{T} = 0.999$; **(3)** PPO with Taylor expansion based advantage estimator, PPO(K). Throughout, we use a single hyper-parameter $\eta = 0.01$. We see that in general, PPO(K) leads to better performance (faster learning speed,

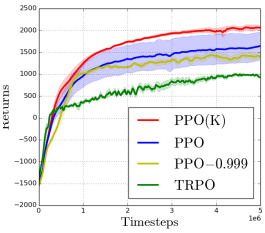


(a) HalfCheetah(G)

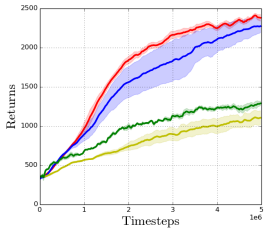
(b) Ant(G)



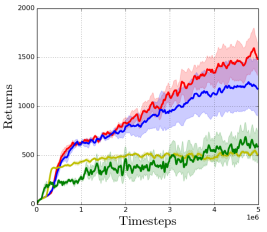
(c) Walker2d(G)



(d) HalfCheetah(B)

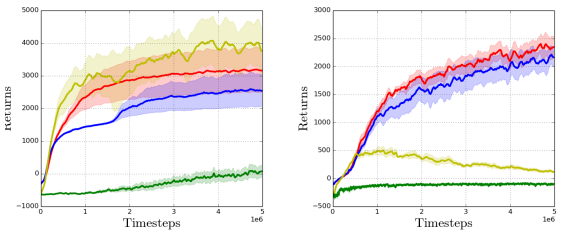


(e) Ant(B)



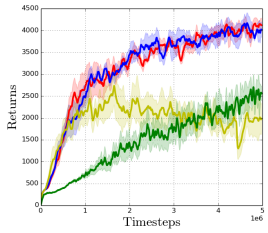
(f) Walker2d(B)

Figure 2. Comparison of Taylor expansion Q-function estimation with other baselines. Each curve shows median ± std results across 5 seeds. Taylor expansion outperforms PPO baselines with both lower and high discount factors.

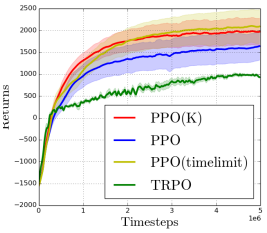


(a) HalfCheetah(G)

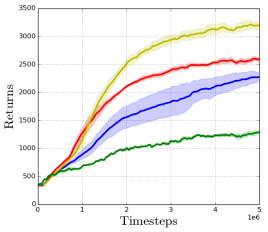
(b) Ant(G)



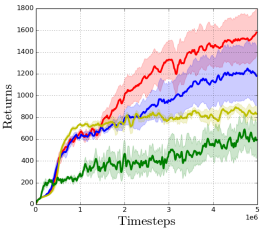
(c) Walker2d(G)



(d) HalfCheetah(B)



(e) Ant(B)



(f) Walker2d(B)

Figure 3. Comparison of Taylor expansion update weighting with other baselines. Each curve shows median ± std results across 5 seeds. Taylor expansion outperforms the default PPO baseline most stably.

better asymptotic performance or smaller variance across 5 seeds). This shows Taylor expansion Q-function estimation could lead to performance gains across tasks, given that the hyper-parameter η is carefully tuned. We provide a detailed ablation study on η in Appendix F, where we show how the overall performance across the benchmark tasks vary as η changes from small to large values.

A second observation is that simply increasing the discount factor to $\gamma = 1 - \frac{1}{T} = 0.999$ generally degrades the performance. This confirms issue with instability of directly applying high discount factors which motivates this work.

We also compare with the open source implementation of (Romoff et al., 2019) in Appendix F, where they estimate $\widehat{Q}^{\pi}_{\gamma'}$ based on recursive bootstraps of Q-function differences. Conceptually, this is similar to Taylor expansions with $K = \infty$. We show that without a careful trade-off mediated by smaller K, this algorithm does not improve performance out of the box.

7.2. Taylor expansion update weighting

As introduced in Section 5, we weigh local gradients $\widehat{Q}_t \nabla_\theta \log \pi_\theta(a_t|x_t)$ with K^{th} order expansion weights $w_{K,\gamma,\gamma'}(t)$. Here, we take $\gamma' = 1 - \frac{1}{T}$. Note that since $K = \infty$ corresponds to $\lim w_{K,\gamma,\gamma'}(t) = (\gamma')^t \approx 1$, this is very close to the commonly implemented PPO baseline. We hence expect the algorithm to work better with relatively large values of K and set $K = 100$ throughout experiments. In practice, we find the performance to be fairly robust in the choice of K. We provide further analysis and ablation study in Appendix F.

Results. We compare a few baselines: **(1)** default PPO; **(2)** PPO with time limit (Pardo et al., 2018). In this case, the states are augmented with time steps $\widetilde{x} \leftarrow [x, t]$ such that the augmented states $\widetilde{x}$ are Markovian; **(3)** PPO with Taylor expansion update weighting PPO(K). In Figure 3, we see that in general, PPO(K) and PPO with time limit outperform the baseline PPO. We speculate that the performance gains

arise from the following empirical motivation: since the evaluation stops at $t = T$, local gradients close to $t = T$ should be weighed down because they do not contribute as much to the final objective. However, the default PPO ignores such an effect and weighs all updates uniformly. To tackle this issue, PPO(K) explicitly weighs down the update while and PPO with time limit augments the state space to restore stationarity. Empirically, though in some cases PPO with time limit also outperforms PPO(K), it behaves fairly unstably in other cases.

Extensions to off-policy algorithms. Above, we mainly focused on on-policy algorithms. The setup is simpler because the data are collected (near) on-policy. It is possible to extend similar results to off-policy algorithms (Mnih et al., 2015; Lillicrap et al., 2015; Fujimoto et al., 2018; Haarnoja et al., 2018). Due to the space limit, we present extended results in Appendix F, where we show how to combine similar techniques to off-policy actor-critic algorithms such as TD3 (Fujimoto et al., 2018) and SAC (Haarnoja et al., 2018) in continuous control domains.

8. Conclusion

We have proposed a family of objectives that interpolate value functions defined with two discount factors. We have shown that similar techniques are applicable to other cumulative quantities defined through discounts, such as PG updates. This framework allowed us to achieve trade-off in estimating value functions or gradient updates, and led to empirical performance gains.

We also highlighted a new direction for bridging the gap between theory and practice: the gap between a fully discounted objective (in theory) and an undiscounted objective (in practice). By building a better understanding of this gap, we shed light on seemingly opaque heuristics which are necessary to achieve good empirical performance. We expect this framework to be useful for new practical algorithms.

Acknowledgements. Yunhao thanks Tadashi Kozuno and Shipra Agrawal for discussions on the discrepancy between policy gradient theory and practices. Yunhao acknowledges the support from Google Cloud Platform for computational resources. The authors also thank Pooria Joulani for reviewing a draft of the paper.

References

Joshua Achiam and OpenAI. Spinning Up in Deep Reinforcement Learning. *https://github.com/openai/spinningup*, 2018.

Ron Amit, Ron Meir, and Kamil Ciosek. Discount factor as a regularizer in reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2020.

Richard Bellman. A Markovian decision process. *Journal of mathematics and mechanics*, pages 679–684, 1957.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI gym. *arXiv*, 2016.

Erwin Coumans. Bullet physics simulation. In *ACM SIGGRAPH 2015 Courses*, page 1. 2015.

Daniela Pucci De Farias and Benjamin Van Roy. The linear programming approach to approximate dynamic programming. *Operations research*, 51(6):850–865, 2003.

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

Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. OpenAI baselines, 2017.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Volodymir Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *Proceeedings of the International Conference on Machine Learning*, 2018.

William Fedus, Carles Gelada, Yoshua Bengio, Marc G Bellemare, and Hugo Larochelle. Hyperbolic discounting and learning over multiple horizons. *arXiv*, 2019.

Roy Fox, Ari Pakman, and Naftali Tishby. Taming the noise in reinforcement learning via soft updates. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2016.

Vincent François-Lavet, Raphael Fonteneau, and Damien Ernst. How to discount deep reinforcement learning: Towards new dynamic strategies. *NIPS Deep Reinforcement Learning Workshop*, 2015.

Scott Fujimoto, Herke Van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the International Conference on Machine Learning*, 2018.

Charles Miller Grinstead and James Laurie Snell. *Introduction to probability*. American Mathematical Soc., 2012.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceeedings of the International Conference on Machine Learning*, 2018.

Hado Hasselt. Double Q-learning. *Advances in Neural Information Processing Systems*, 2010.

Michael Janner, Igor Mordatch, and Sergey Levine. Gamma-models: Generative temporal difference learning for infinite-horizon prediction. *Advances in Neural Information Processing Systems*, 2020.

Nan Jiang, Satinder P Singh, and Ambuj Tewari. On structural properties of MDPs that bound loss due to shallow planning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2016.

Steven Kapturowski, Georg Ostrovski, John Quan, Remi Munos, and Will Dabney. Recurrent experience replay in distributed reinforcement learning. In *Proceedings of the International Conference on Learning Representations*, 2018.

Michael J Kearns and Satinder P Singh. Bias-variance error bounds for temporal difference updates. In *Proceedings of the Conference on Learning Theory*, 2000.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Romain Laroche and Harm van Seijen. In reinforcement learning, all objective functions are not equal. 2018.

Lucas Lehnert, Romain Laroche, and Harm van Seijen. On value function representation of long horizon problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *Proceedings of the International Conference on Learning Representations*, 2015.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

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

Chris Nota and Philip S Thomas. Is the policy gradient a gradient? In *Proceedings of the International Conference on Autonomous Agenst and Multiagent Systems*, 2019.

Brendan O'Donoghue, Remi Munos, Koray Kavukcuoglu, and Volodymyr Mnih. Combining policy gradient and Q-learning. In *Proceedings of the International Conference on Learning Representations*, 2016.

Fabio Pardo, Arash Tavakoli, Vitaly Levdik, and Petar Kormushev. Time limits in reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2018.

Tobias Pohlen, Bilal Piot, Todd Hester, Mohammad Gheshlaghi Azar, Dan Horgan, David Budden, Gabriel Barth-Maron, Hado Van Hasselt, John Quan, Mel Večerík, Matteo Hessel, Rémi Munos, and Olivier Pietquin. Observe and look further: Achieving consistent performance on atari. *arXiv*, 2018.

Danil V Prokhorov and Donald C Wunsch. Adaptive critic designs. *IEEE transactions on Neural Networks*, 8(5): 997–1007, 1997.

Martin L Puterman. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Joshua Romoff, Peter Henderson, Ahmed Touati, Emma Brunskill, Joelle Pineau, and Yann Ollivier. Separating value functions across time-scales. *Proceedings of the International Conference on Machine Learning*, 2019.

Sheldon M Ross. *Introduction to probability models*. Academic press, 2014.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *Proceedings of the International Conference on Learning Representations*, 2015.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the International Conference on Machine Learning*, 2015a.

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

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv*, 2017.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic

policy gradient algorithms. In *Proceedings of the International Conference on Machine Learning*, 2014.

Samarth Sinha, Jiaming Song, Animesh Garg, and Stefano Ermon. Experience replay with likelihood-free importance weights. *arXiv*, 2020.

Richard S Sutton. TD models: Modeling the world at a mixture of time scales. In *Proceedings of the International Conference on Machine Learning*. 1995.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT Press, 2018.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, 2000.

Yunhao Tang, Michal Valko, and Rémi Munos. Taylor expansion policy optimization. In *Proceedings of the International Conference on Machine Learning*, 2020.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *International Conference on Intelligent Robots and Systems*, 2012.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double Q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Hado van Hasselt, John Quan, Matteo Hessel, Zhongwen Xu, Diana Borsa, and André Barreto. General non-linear Bellman equations. *arXiv*, 2019.

Harm Van Seijen, Mehdi Fatemi, and Arash Tavakoli. Using a logarithmic mapping to enable lower discount factors in reinforcement learning. In *Advances in Neural Information Processing Systems*, 2019.

Zhongwen Xu, Hado P van Hasselt, and David Silver. Meta-gradient reinforcement learning. *Advances in Neural Information Processing Systems*, 2018.

Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2008.

APPENDICES: Taylor Expansions of Discount Factors

A. Proofs

Proposition 3.1. The following holds for all $K \geq 0$,

$$V_{\gamma'}^{\pi} = \sum_{k=0}^{K} \left((\gamma' - \gamma)(I - \gamma P^{\pi})^{-1} P^{\pi}\right)^{k} V_{\gamma}^{\pi} + \underbrace{\left((\gamma' - \gamma)(I - \gamma P^{\pi})^{-1} P^{\pi}\right)^{K+1} V_{\gamma'}^{\pi}}_{\text{residual}}. \tag{9}$$

When $\gamma < \gamma' < 1$, the residual norm converges to 0, which implies

$$V_{\gamma'}^{\pi} = \sum_{k=0}^{\infty} \left((\gamma' - \gamma)(I - \gamma P^{\pi})^{-1} P^{\pi}\right)^{k} V_{\gamma}^{\pi}. \tag{10}$$

Proof. Recall the Woodbury matrix identity

$$(I - \gamma' P^{\pi})^{-1} = (I - \gamma P^{\pi})^{-1} + (\gamma' - \gamma)(I - \gamma P^{\pi})^{-1} P^{\pi} (I - \gamma' P^{\pi})^{-1}.$$

Recall the equality $V_{\gamma'}^{\pi} = (I - \gamma' P^{\pi})^{-1} r^{\pi}$. By plugging in the Woodbury matrix identity, this immediate shows

$$\begin{aligned} V_{\gamma'}^{\pi} &= (I - \gamma P^{\pi})^{-1} r^{\pi} + (\gamma' - \gamma)(I - \gamma P^{\pi})^{-1} P^{\pi} (I - \gamma' P^{\pi})^{-1} r^{\pi} \\ &= V_{\gamma}^{\pi} + (\gamma' - \gamma)(I - \gamma P^{\pi})^{-1} P^{\pi} V_{\gamma'}^{\pi}. \end{aligned}$$

Now, observe that the second term involves $V_{\gamma'}^{\pi}$. We can plug in the definition of $V_{\gamma'}^{\pi} = (I - \gamma' P^{\pi})^{-1} r^{\pi}$ and invoke the Woodbury matrix identity again. This produces

$$V_{\gamma'}^{\pi} = V_{\gamma}^{\pi} + (\gamma' - \gamma)(I - \gamma P^{\pi})^{-1} P^{\pi} V_{\gamma}^{\pi} + \left((\gamma' - \gamma)(I - \gamma P^{\pi})^{-1} P^{\pi}\right)^{2} V_{\gamma'}^{\pi}.$$

By induction, it is straightforward to show that iterating the above procedure $K \geq 0$ times produces the following equalities

$$V_{\gamma'}^{\pi} = \sum_{k=0}^{K} \left((\gamma' - \gamma)(I - \gamma P^{\pi})^{-1} P^{\pi}\right)^{k} V_{\gamma}^{\pi} + + \underbrace{\left((\gamma' - \gamma)(I - \gamma P^{\pi})^{-1} P^{\pi}\right)^{K+1} V_{\gamma'}^{\pi}}_{\text{residual}}.$$

Consider the norm of the residual term. Since P^{π} is a transition matrix, $\|P^{\pi}\|_{\infty} < 1$. As a result, $\left\|(I - \gamma P^{\pi})^{-1}\right\|_{\infty} = \|\sum_{t=0}^{\infty} \gamma^{t} (P^{\pi})^{t}\|_{\infty} < (1 - \gamma)^{-1}$. This implies

$$\left\|\left((\gamma' - \gamma)(I - \gamma P^{\pi})^{-1} P^{\pi}\right)^{K+1} V_{\gamma'}^{\pi}\right\|_{\infty} < \left(\frac{\gamma' - \gamma}{1 - \gamma}\right)^{K+1} \cdot \frac{R_{\max}}{1 - \gamma'}.$$

When $\gamma < \gamma' < 1$, the residual norm decays exponentially and $\to 0$ as $K \to \infty$. This implies that the infinite series converges,

$$V_{\gamma'}^{\pi} = \sum_{k=0}^{\infty} \left((\gamma' - \gamma)(I - \gamma P^{\pi})^{-1} P^{\pi}\right)^{k} V_{\gamma}^{\pi}.$$

Additional consideration when $\gamma' = 1$. When $\gamma' = 1$, in order to ensure finiteness of $V_{\gamma'=1}^{\pi}$, we assume the following two conditions: **(1)** The Markov chain induced by π is absorbing; **(2)** for any absorbing state x, $r^{\pi}(x) = 0$. Without loss of generality, assume there exists a single absorbing state. In general, the transition matrix P^{π} can be decomposed as follows (Grinstead and Snell, 2012; Ross, 2014),

$$P^{\pi} = \begin{pmatrix} \widetilde{P}^{\pi} & \widetilde{p}^{\pi} \\ 0 & 1 \end{pmatrix},$$

where $\widetilde{P}^{\pi} \in \mathbb{R}^{(|\mathcal{X}|-1)|\mathcal{A}| \times (|\mathcal{X}|-1)|\mathcal{A}|}$ and $\widetilde{p}^{\pi} \in \mathbb{R}^{\mathcal{X}-1)}$. Here, the first $\mathcal{X} - 1$ states are transient and the last state is absorbing. For convenience, define $\widetilde{r}^{\pi}$ as the reward vector r^{π} constrained on the first $\mathcal{X} - 1$ transient states. We provide a few lemmas below.

Lemma A.1. The matrix $(I - \widetilde{P})^\pi$ is invertible and its inverse is $(I - \widetilde{P})^\pi = \sum_{k=0}^{\infty} (\widetilde{P})^k$.

Proof. Define a matrix $N = \sum_{k=0}^{\infty} (\widetilde{P}^\pi)^k$, then $N[x, y]$ defines the expected number of times it takes to transition from x to y before absorption. By definition of the absorbing chain, N is finite. This further shows that $(I - \widetilde{P}^\pi)$ is invertible, because

$$N(I - \widetilde{P}^\pi) = (I - \widetilde{P}^\pi)N = I.$$

□

Lemma A.2. Let $f(A, B)$ be a matrix polynomial function of matrix A and B. Then

$$f\left(P^\pi, (I - \gamma P^\pi)^{-1}\right) = \begin{pmatrix} f\left(\widetilde{P}^\pi, \left(I - \gamma \widetilde{P}^\pi\right)^{-1}\right) & B \\ 0 & 1 \end{pmatrix},$$

where B is some matrix.

Proof. The intuition for the above result is that polynomial transformation preserves the *block triangle* property of P^π and $(I - \gamma P^\pi)^{-1}$. In general, we can assume

$$f(A, B) = \sum_{m,n \le K} c_{m,n} A^m B^n,$$

for some $K \ge 0$ and $c_{m,n} \in \mathbb{R}$ are scalar coefficients. First, note that $(P^\pi)^k, k \ge 0$ is of the form

$$(P^\pi)^k = \begin{pmatrix} \left(\widetilde{P}^\pi\right)^k & C \\ 0 & 1 \end{pmatrix},$$

for some matrix C. Since $(I - \gamma A)^{-1} = \sum_{k=0}^{\infty} A^k$ for $A \in \{P^\pi, \widetilde{P}^\pi\}$, this implies that

$$(I - \gamma P^\pi)^{-1} = \sum_{k=0}^{\infty} (\gamma P^\pi)^k = \begin{pmatrix} \left(\widetilde{P}^\pi\right)^k & D \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \left(I - \gamma \widetilde{P}^\pi\right)^{-1} & D \\ 0 & 1 \end{pmatrix},$$

for some matrix D. The above two results show that both polynomials of P^π and $(I - \gamma P^\pi)^{-1}$ are *block upper triangle* matrices. It is then straightforward that

$$(P^\pi)^m \left((I - \gamma P^\pi)^{-1}\right)^n = \begin{pmatrix} \left(\widetilde{P}^\pi\right)^m \left(\left(I - \gamma \widetilde{P}^\pi\right)^{-1}\right)^n & E \\ 0 & 1 \end{pmatrix},$$

for some matrix E. Finally, since $f(P^\pi, (I - \gamma P^\pi)^{-1})$ is a linear combination of $(P^\pi)^m \left((I - \gamma P^\pi)^{-1}\right)^n$, we conclude the proof. □

Lemma A.3. Under assumption **(1)** and **(2)**, one could write the value function $V^\pi_{\gamma'=1}$ as

$$V^\pi_{\gamma'=1} = \sum_{k=0}^{\infty} (P^\pi)^t r^\pi,$$

where the infinite series on the RHS converges. In addition, for any transient state x, $V^\pi_{\gamma'=1}(x) = \left[\sum_{k=0}^{\infty} (\widetilde{P}^\pi)^k \widetilde{r}^\pi\right](x)$.

Proof. Recall that $V^\pi_\gamma(x) \coloneqq \mathbb{E}\left[\sum_{t=0}^\infty r_t \mid x_0 = x\right]$. Under assumption **(2)**, for any absorbing state x, $V^\pi_\gamma(x) = 0 = \left[\sum_{k=0}^\infty (P^\pi)^k r^\pi\right](x)$. We can instead constrain the Markov chain to the transient states. For any transient state x, recall the definition of N from Lemma A.1, it follows that

$$V^\pi_{\gamma'=1}(x) = \sum_y N(\text{expected number of times in } y | x_0 = x) r^\pi(y) = \left[N r^\pi\right](x) = \left[\left(I - \widetilde{P}^\pi\right)^{-1} \widetilde{r}^\pi\right](x) = \left[\sum_{k=0}^\infty (\widetilde{P}^\pi)^k \widetilde{r}^\pi\right](x).$$

By Lemma A.2, this is equivalent to $\left[\sum_{k=0}^\infty (P^\pi)^t r^\pi\right](x)$. We thus complete the proof. □

Lemma A.4. The following holds for any $\gamma < 1$,

$$\begin{aligned}
\left(I - \widetilde{P}^\pi\right)^{-1} &= \left(I - \gamma\widetilde{P}^\pi - (1-\gamma)\widetilde{P}^\pi\right)^{-1} \\
&= \sum_{k=0}^K \left((1-\gamma)\left(I - \gamma\widetilde{P}^\pi\right)^{-1}\widetilde{P}^\pi\right)^k \left(I - \gamma\widetilde{P}^\pi\right)^{-1} + \left((1-\gamma)\left(I - \gamma\widetilde{P}^\pi\right)^{-1}\widetilde{P}^\pi\right)^{K+1}\left(I - \widetilde{P}^\pi\right)^{-1} \\
&= \sum_{k=0}^\infty \left((1-\gamma)\left(I - \gamma\widetilde{P}^\pi\right)^{-1}\widetilde{P}^\pi\right)^k \left(I - \gamma\widetilde{P}^\pi\right)^{-1}. \qquad (20)
\end{aligned}$$

Proof. The first two lines derive from a straightforward application of Woodbury matrix identity to $(I - \widetilde{P}^\pi)^{-1}$. This is valid because by Lemma A.1, $(I - \widetilde{P}^\pi)$ is invertible. The convergence of the infinite series is guaranteed for all $\gamma < 1$. To see why, recall that the finiteness of $N = \sum_{k=0}^\infty (\widetilde{P}^\pi)^k$ implies $(\widetilde{P}^\pi)^{K+1} \to 0$. We can bound the residual,

$$\left\|\left((1-\gamma)\left(I - \gamma\widetilde{P}^\pi\right)^{-1}\widetilde{P}^\pi\right)^{K+1}\left(I - \widetilde{P}^\pi\right)^{-1}\right\|_\infty \le \left\|(\widetilde{P}^\pi)^{K+1}\left(I - \widetilde{P}^\pi\right)^{-1}\right\|_\infty \to 0.$$

□

Finally, we combine results from the above to prove the main claim. First, consider the absorbing state x. Due to Assumption **(2)**, $V^\pi_\gamma(x) = 0$ for any $\gamma \in [0, 1]$. The matrix equalities in Proposition 3.2 holds in this case.

In the following, we consider any transient states x. By Lemma A.3 and Lemma A.4

$$\begin{aligned}
\widetilde{V}^\pi_{\gamma'=1}(x) &= \left[\sum_{k=0}^\infty (\widetilde{P}^\pi)^k \widetilde{r}^\pi\right](x) \\
&= \left[\sum_{k=0}^K \left((1-\gamma)\left(I - \gamma\widetilde{P}^\pi\right)^{-1}\widetilde{P}^\pi\right)^k \left(I - \gamma\widetilde{P}^\pi\right)^{-1}\widetilde{r}^\pi + \left((1-\gamma)\left(I - \gamma\widetilde{P}^\pi\right)^{-1}\widetilde{P}^\pi\right)^{K+1}\left(I - \widetilde{P}^\pi\right)^{-1}\widetilde{r}^\pi\right](x)
\end{aligned}$$

Now, notice that because the last entries of $r^\pi, V^\pi_\gamma, V^\pi_{\gamma'=1}$ are zero (for the absorbing state),

$$\left[\left(I - \gamma\widetilde{P}^\pi\right)^{-1}\widetilde{r}^\pi\right](x) = \left[\left(I - \gamma P^\pi\right)^{-1} r^\pi\right](x).$$

Combining with Lemma A.2,

$$\begin{aligned}
\widetilde{V}^\pi_{\gamma'=1}(x) &= \left[\sum_{k=0}^K \left((1-\gamma)\left(I - \gamma P^\pi\right)^{-1} P^\pi\right)^k \left(I - \gamma P^\pi\right)^{-1}\widetilde{r}^\pi + \left((1-\gamma)\left(I - \gamma P^\pi\right)^{-1} P^\pi\right)^{K+1} V^\pi_{\gamma'=1}\right](x) \\
&= \left[\underbrace{\sum_{k=0}^K \left((1-\gamma)\left(I - \gamma P^\pi\right)^{-1} P^\pi\right)^k V^\pi_\gamma}_{\text{K-th order expansion}} + \underbrace{\left((1-\gamma)\left(I - \gamma P^\pi\right)^{-1} P^\pi\right)^{K+1} V^\pi_{\gamma'=1}}_{\text{residual}}\right](x).
\end{aligned}$$

The residual term $\to 0$ as $K \to 0$ with similar arguments used for Lemma A.4. We hence conclude the proof. □

Proposition 3.2. The following bound holds for all $K \geq 0$,

$$\left|V^\pi_{\gamma'}(x) - V^\pi_{K,\gamma,\gamma'}(x)\right| \leq \left(\frac{\gamma' - \gamma}{1-\gamma}\right)^{K+1} \frac{R_{\max}}{1-\gamma'}. \tag{12}$$

Proof. The proof follows directly from the residual term in Proposition 3.1. Recall that the residual term takes the form

$$V^\pi_{\gamma'} - V^\pi_{K,\gamma,\gamma'} = \left((\gamma' - \gamma)(I - \gamma P^\pi)^{-1} P^\pi\right)^{K+1} V^\pi_{\gamma'}.$$

Its infinity norm can be bounded as $(\frac{\gamma'-\gamma}{1-\gamma})^{K+1} \frac{R_{\max}}{1-\gamma'}$ □

Lemma 4.1. Assume $\gamma < \gamma' < 1$. We can write $V^\pi_{\gamma'}(x) = (\rho^\pi_{x,\gamma,\gamma'})^T V^\pi_\gamma$, where the weight vector $\rho^\pi_{x,\gamma,\gamma'} \in \mathbb{R}^{\mathcal{X}}$ is

$$\left(I - \gamma (P^\pi)^T\right)\left(I - \gamma' (P^\pi)^T\right)^{-1} \delta_x.$$

Also we can rewrite $V^\pi_{\gamma'}(x)$, using an expectation, as:

$$V^\pi_\gamma(x) + \mathbb{E}_\pi\left[\sum_{t=1}^{\infty} (\gamma' - \gamma)(\gamma')^{t-1} V^\pi_\gamma(x_t) \,\middle|\, x_0 = x\right]. \tag{16}$$

When $\gamma' = 1$, $\rho^\pi_{x,\gamma,\gamma'}$ might be undefined. However, Eqn (16) still holds if assumptions **A.1** and **A.2** are satisfied.

Proof. We will derive the above result with the matrix form. Recall by applying Woodbury inversion identity to $(I - \gamma' P^\pi)^{-1} = (I - (\gamma' - \gamma)P^\pi - \gamma P^\pi)^{-1}$, we get

$$\begin{aligned}
(I - \gamma' P^\pi)^{-1} &= \sum_{k=0}^{\infty} \left((\gamma' - \gamma)(I - \gamma P^\pi)^{-1} P^\pi\right)^k (I - \gamma P^\pi)^{-1} \\
&= (I - \gamma P^\pi)^{-1} + \sum_{k=1}^{\infty} \left((\gamma' - \gamma)(I - \gamma P^\pi)^{-1} P^\pi\right)^k (I - \gamma P^\pi)^{-1} \\
&= (I - \gamma P^\pi)^{-1} + (\gamma' - \gamma) \sum_{k=1}^{\infty} \left((\gamma' - \gamma)(I - \gamma P^\pi)^{-1} P^\pi\right)^k \cdot (I - \gamma P^\pi)^{-1} \cdot P^\pi (I - \gamma P^\pi)^{-1} \\
&= (I - \gamma P^\pi)^{-1} + (\gamma' - \gamma)(I - \gamma' P^\pi)^{-1} \cdot P^\pi \cdot (I - \gamma P^\pi)^{-1}.
\end{aligned}$$

Then, right multiply the above equation by r^π,

$$\begin{aligned}
V^\pi_{\gamma'} &= V^\pi_\gamma + (\gamma' - \gamma)(I - \gamma' P^\pi)^{-1} P^\pi V^\pi_\gamma \\
&= V^\pi_\gamma + (\gamma' - \gamma) \sum_{t=1}^{\infty} (\gamma')^{t-1} (P^\pi)^t V^\pi_\gamma.
\end{aligned}$$

By indexing both sides at state x, we recover the following equality,

$$V^\pi_{\gamma'}(x) = V^\pi_\gamma(x) + \mathbb{E}_\pi\left[\sum_{t=1}^{\infty} (\gamma' - \gamma)(\gamma')^{t-1} V^\pi_\gamma(x_t) \,\middle|\, x_0 = x\right].$$

To derive the expression for $\rho^\pi_{x,\gamma,\gamma}$, note that also

$$V^\pi_{\gamma'} = (I - \gamma' P^\pi)^{-1} r^\pi = (I - \gamma P^\pi)(I - \gamma' P^\pi)^{-1}(I - \gamma P^\pi)^{-1} r^\pi = \underbrace{(I - \gamma P^\pi)(I - \gamma' P^\pi)^{-1}}_{\text{weight matrix } W} V^\pi_\gamma,$$

where we use the fact that $(I - \gamma P^\pi)$ commutes with $(I - \gamma' P^\pi)^{-1}$. Since we define $\rho^\pi_{x,\gamma,\gamma'}$ as such that $V^\pi_{\gamma'}(x) = \left(\rho^\pi_{x,\gamma,\gamma'}\right)^T V^\pi_\gamma$, we can derive the matrix form of $\rho^\pi_{x,\gamma,\gamma'}$ by indexing the x-th row of weight matrix W. This directly leads to the desired result

$$\rho^\pi_{x,\gamma,\gamma'} = \left(I - \gamma (P^\pi)^T\right)\left(I - \gamma' (P^\pi)^T\right)^{-1} \delta_x.$$

□

Proposition 4.3. For any $\gamma < \gamma' < 1$, the first partial gradient $(\partial_V F(V_\gamma^{\pi_\theta}, \rho_{x,\gamma,\gamma'}^{\pi_\theta}))^T \nabla_\theta V_\gamma^{\pi_\theta}$ can be expressed as

$$\mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty}(\gamma')^t Q_\gamma^{\pi_\theta}(x_t,a_t)\nabla_\theta \log \pi_\theta(a_t|x_t) \;\middle|\; x_0 = x\right]. \tag{17}$$

When $\gamma' = 1$, under assumptions **A.1** and **A.2**, the first partial gradient exists and is expressed as

$$\mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T} Q_\gamma^{\pi_\theta}(x_t,a_t)\nabla_\theta \log \pi_\theta(a_t|x_t) \;\middle|\; x_0 = x\right]. \tag{18}$$

Proof. First we assume $\gamma' < 1$, we will consider the extension to $\gamma' = 1$ at the end of the proof. Recall that the policy gradient takes the following form,

$$\nabla_\theta V_\gamma^{\pi_\theta}(x) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t Q_\gamma^{\pi_\theta}(x_t,a_t)\nabla_\theta \log \pi_\theta(a_t|x_t) \;\middle|\; x_0 = x.\right]$$

We plug in the above, the partial derivative $\left(\partial_V F(V_\gamma^{\pi_\theta}, \rho_{x,\gamma,\gamma'}^{\pi_\theta})\right)^T \nabla_\theta V_\gamma^{\pi_\theta}$ evaluates to the following

$$\begin{aligned}
&\nabla_\theta V_\gamma^{\pi_\theta}(x) + \mathbb{E}_{\pi_\theta}\left[(\gamma'-\gamma)\sum_{t=1}^{\infty}(\gamma')^{t-1}\nabla_\theta V_\gamma^{\pi_\theta}(x_t)\right] \\
&= \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t Q_\gamma^{\pi_\theta}(x_t,a_t)\nabla_\theta \log \pi_\theta(a_t|x_t) \;\middle|\; x_0 = x\right] \\
&+ \mathbb{E}_{\pi_\theta}\left[(\gamma'-\gamma)(\gamma')^{t-1}\sum_{t=1}^{\infty}\sum_{s=0}^{\infty}\gamma^s Q_\gamma^{\pi_\theta}(x_{t+s},a_{t+s})\nabla_\theta \log \pi_\theta(a_{t+s}|x_{t+s}) \;\middle|\; x_0 = x\right] \\
&= \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty}\left(\underbrace{\gamma^t + \sum_{u=1}^{t}(\gamma'-\gamma)(\gamma')^{u-1}\gamma^{t-u}}_{\text{coefficient at time t}}\right) Q_\gamma^{\pi_\theta}(x_t,a_t)\nabla_\theta \log \pi_\theta(a_t|x_t) \;\middle|\; x_0 = x.\right]
\end{aligned}$$

In the above, the coefficient term at time t can be calculated by carefully grouping terms across different time steps. It can be shown that the coefficient term evaluates to $(\gamma')^t$ for all $t \geq 0$. This concludes the proof.

Alternative proof based on matrix notations. We introduce an alternative proof based on matrix notations as it will make the extension to $\gamma' = 1$ simpler. First, note that

$$V_{\gamma'}^{\pi} = \left(I - \gamma' P^{\pi}\right)^{-1} r^{\pi} = \left(I - \gamma' P^{\pi}\right)^{-1}\left(I - \gamma P^{\pi}\right)\left(I - \gamma P^{\pi}\right)^{-1} r^{\pi} = \left(I - \gamma P^{\pi}\right)\left(I - \gamma' P^{\pi}\right)^{-1}\left(I - \gamma P^{\pi}\right)^{-1} r^{\pi},$$

where for the second equality we exploit the fact that $(I - \gamma P^{\pi})$ commutes with $(I - \gamma' P^{\pi})^{-1}$. Now, notice that the above rewrites as

$$V_{\gamma'}^{\pi} = \underbrace{\left(I - \gamma P^{\pi}\right)\left(I - \gamma' P^{\pi}\right)^{-1}}_{W_{\gamma,\gamma'}} V_{\gamma}^{\pi}$$

where $W_{\gamma,\gamma'}$ is the weight matrix. This matrix is equivalent to the weighting distribution $\rho_{x,\gamma,\gamma'}^{\pi}$ by $W_{\gamma,\gamma'}[x] = \rho_{x,\gamma,\gamma'}^{\pi}$ where $A[x]$ is the x-th row of matrix A. The first partial gradient corresponds to differentiating $V_{\gamma'}^{\pi_\theta}$ only through $V_{\gamma}^{\pi_\theta}$. To make the derivation clear in matrix notations, let θ_i be the i-th component of the parameter θ. Define $\nabla_{\theta_i} V_\gamma^{\pi_\theta} \in \mathbb{R}^{\mathcal{X}}$ such that $\nabla_{\theta_i} V_\gamma^{\pi_\theta}(x) = \nabla_{\theta_i} V_\gamma^{\pi_\theta}(x)$, This means the i-th component of the first partial gradient across all states is

$$W_{\gamma,\gamma'} \nabla_{\theta_i} V_\gamma^{\pi_\theta} \in \mathbb{R}^{\mathcal{X}}.$$

Let $G_\gamma^{\theta_i} \in \mathbb{R}^{\mathcal{X}}$ to be the vector of local gradient (for parameter θ_i) such that $G_\gamma^{\theta_i}(x) = \sum_a \nabla_{\theta_i}\pi_\theta(a|x)Q_\gamma^{\pi_\theta}(x,a)$. Vanilla PG (Sutton et al., 2000) can be expressed as

$$\nabla_{\theta_i} V_\gamma^{\pi_\theta} = (I - \gamma P^\pi)^{-1} G_\gamma^{\theta_i}.$$

We can finally derive the following,

$$\begin{aligned} W_{\gamma,\gamma'}\nabla_{\theta_i}V_\gamma^{\pi_\theta} &= (I-\gamma P^\pi)(I-\gamma' P^\pi)^{-1}G_\gamma^{\theta_i} \\ &= (I-\gamma P^\pi)(I-\gamma' P^\pi)^{-1}(I-\gamma P^\pi)^{-1}G_\gamma^{\theta_i} \\ &= (I-\gamma P^\pi)(I-\gamma P^\pi)^{-1}(I-\gamma' P^\pi)^{-1}G_\gamma^{\theta_i} \\ &= (I-\gamma' P^\pi)^{-1}G_\gamma^{\theta_i} \end{aligned}$$

Now, consider the x-th component of the above vector. We have $\nabla_\theta[J(\pi_\theta,\pi_t)]_{\pi_t=\pi_\theta}$ is equal to

$$\mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty}(\gamma')^t\sum_a \nabla_{\theta_i}\pi_\theta(a|x_t)Q_\gamma^{\pi_\theta}(x_t,a)\,\middle|\, x_0=x\right] = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty}(\gamma')^t\nabla_{\theta_i}\log\pi_\theta(a_t|x_t)Q_\gamma^{\pi_\theta}(x_t,a)\,\middle|\, x_0=x\right]$$

When concatenating the gradient for all component $\theta_i i$ of θ, we conclude the proof.

Extensions to the case $\gamma' = 1$. Similar to the arguments made in the proof of Proposition 3.2, under assumptions **A.1** and **A.2**, we can decompose the transition matrix P^π as

$$P^\pi = \begin{pmatrix} \widetilde{P} & \widetilde{p} \\ 0 & 1 \end{pmatrix},$$

where the last state is assumed to be absorbing. Though $(I-\gamma' P^\pi)^{-1}$ for $\gamma'=1$ is in general not necessarily invertible, the matrix $(I-\widetilde{P})^{-1}$ is invertible. Since $r^\pi(x)$ for the absorbing state x, we have deduced that $Q_\gamma^\pi(x.a) = V_\gamma^\pi(x) = 0$, and accordingly $G_\gamma^{\theta_i}(x) = 0$. As such, though $(I-\gamma' P^\pi)^{-1}$ for $\gamma'=1$ might be undefined, the multiplication $(I-\gamma' P^\pi)^{-1}G_\gamma^{\theta_i}$ is defined, with the last entry being 0. Since at time $t=T$, the chain enters the absorbing states, all local gradient terms that come after T are zero. As a result, the x-th component of $(I-\gamma' P^\pi)^{-1}G_\gamma^{\theta_i}$ is

$$\mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T}(\gamma')^t\nabla_{\theta_i}\log\pi_\theta(a_t|x_t)Q_\gamma^{\pi_\theta}(x_t,a)\,\middle|\, x_0=x\right]$$

□

Proposition 4.4. Assume $\gamma < \gamma' < 1$. For any $x \in \mathcal{X}$, define the K^{th} Taylor expansion to $\rho^\pi_{x,\gamma,\gamma'}$ as

$$\rho^\pi_{x,K,\gamma,\gamma'} = \sum_{k=0}^{K}\left((\gamma'-\gamma)\left(I-\gamma(P^\pi)^T\right)^{-1}(P^\pi)^T\right)^k\delta_x.$$

It can be shown that $V^\pi_{K,\gamma,\gamma'}(x) = (\rho^\pi_{x,K,\gamma,\gamma'})^T V_\gamma^\pi$ and $\left\|\rho^\pi_{x,K,\gamma,\gamma'} - \rho^\pi_{K,\gamma,\gamma'}\right\|_\infty = O\left(\left(\frac{\gamma'-\gamma}{1-\gamma}\right)^{K+1}\right)$.

Proof. Recall from Lemma 4.1, by construction,

$$\rho^\pi_{x,\gamma,\gamma'} = \left(I-\gamma'(P^\pi)^T\right)^{-1}\left(I-\gamma(P^\pi)^T\right)\delta_x.$$

Simialr to the case of primal space expansions in Section 3.1, we construct the K^{th} order expansion to $\rho^\pi_{x,\gamma,\gamma'}$ via the expansion of the matrix $\left(I-\gamma(P^\pi)^T\right)^{-1}$. Recall that

$$\left(I-\gamma'(P^\pi)^T\right)^{-1} = \sum_{k=0}^{\infty}\left((\gamma'-\gamma)(P^\pi)^T\left(I-\gamma(P^\pi)^T\right)^{-1}\right)^k\left(I-\gamma(P^\pi)^T\right)^{-1}.$$

□

When truncating the infinite series to the first $K+1$ terms, we derive the K^{th} order expansion $\rho^{\pi}_{x,K,\gamma,\gamma'}$,

$$\left((\gamma'-\gamma)(P^\pi)^T\left(I-\gamma(P^\pi)^T\right)^{-1}\right)^k\left(I-\gamma(P^\pi)^T\right)^{-1}\left(I-\gamma(P^\pi)^T\right)\delta_x=\sum_{k=0}^{K}\left((\gamma'-\gamma)(P^\pi)^T\left(I-\gamma(P^\pi)^T\right)^{-1}\right)^k\delta_x.$$

Note that since

$$\left\|\sum_{k=K+1}^{\infty}\left((\gamma'-\gamma)(P^\pi)^T\left(I-\gamma(P^\pi)^T\right)^{-1}\right)^k\left(I-\gamma(P^\pi)^T\right)^{-1}\right\|_\infty\leq\left(\frac{\gamma'-\gamma}{1-\gamma}\right)^{K+1}\frac{1}{1-\gamma'}.$$

This concludes the proof.

B. Further results on Taylor expansions in the dual space

The dual representation of value function $V^\pi_{\gamma'}(x)$ in Eqn (6) is $V^\pi_{\gamma'}(x)=(1-\gamma')^{-1}(r^\pi)^T d^\pi_{x,\gamma'}$ where $r^\pi, d^\pi_{x,\gamma'}\in\mathbb{R}^{\mathcal{X}}$ are vector rewards and visitation distribution starting at state x. Here, we abuse the notation $d^\pi_{x,\gamma}$ to denote both a function and a vector, i.e., $d^\pi_{x,\gamma}(x')$ can be interpreted as both a function evaluation and a vector indexing. Given such a dual representation, one natural question is whether the K^{th} expansion in the primal space corresponds to some approximations of the discounted visitation distribution $d^\pi_{K,\gamma,\gamma'}\approx d^\pi_{x,\gamma'}$. Below, we answer in the affirmative.

Let $\delta_x\in\mathbb{R}^{\mathcal{X}}$ be the one-hot distribution such that $[\delta_x]_{x'}=1$ only when $x'=x$. The visitation distribution satisfies the following balance equation in matrix form

$$d^\pi_{x,\gamma'}=(1-\gamma')\delta_x+\gamma'(P^\pi)^T d^\pi_{x,\gamma'}. \tag{21}$$

Inverting the equation, we obtain an explicit expression for the visitation distribution $d^\pi_{x,\gamma'}=(1-\gamma')(I-\gamma' P^\pi)^{-1}\delta_x$. Following techniques used in the derivation of Propo 3.1, we can derive similar approximation results for dual variables. See Appendix B.

Proposition B.1. The following holds for all $K\geq 0$,

$$\begin{aligned}d^\pi_{x,\gamma'}=&\frac{1-\gamma'}{1-\gamma}\sum_{k=0}^{K}\left((\gamma'-\gamma)\left(I-\gamma(P^\pi)^T\right)^{-1}(P^\pi)^T\right)^k d^\pi_{x,\gamma}.\\&+\underbrace{\left((\gamma'-\gamma)\left(I-\gamma(P^\pi)^T\right)^{-1}(P^\pi)^T\right)^{K+1}d^\pi_{x,\gamma'}}_{\text{residual}}.\end{aligned} \tag{22}$$

When $\gamma<\gamma'<1$, the residual norm $\to 0$, which implies that the following holds

$$d^\pi_{x,\gamma'}=\frac{1-\gamma'}{1-\gamma}\sum_{k=0}^{\infty}((\gamma'-\gamma)\left(I-\gamma(P^\pi)^T\right)^{-1}(P^\pi)^T)^k d^\pi_{x,\gamma}. \tag{23}$$

Proof. Starting from the fixed point equation satisfied by $d^\pi_{\gamma'}$, we can apply Woodbury inversion indentity

$$\begin{aligned}d^\pi_{\gamma'}&=(1-\gamma')\left(I-\gamma'(P^\pi)^T\right)^{-1}\delta_x\\&=(1-\gamma')\sum_{k=0}^{K}\left((\gamma'-\gamma)(I-\gamma P^\pi)^{-1}(P^\pi)^T\right)^k\delta_x+(1-\gamma')\left((\gamma'-\gamma)\left(I-\gamma(P^\pi)^T\right)^{-1}(P^\pi)^T\right)^K\left(I-\gamma'(P^\pi)^T\right)^{-1}\delta_x\\&=\frac{1-\gamma'}{1-\gamma}\sum_{k=0}^{K}\left((\gamma'-\gamma)\left(I-\gamma(P^\pi)\right)^{-1}P^\pi\right)^k d^\pi_{\gamma}+(1-\gamma')\left((\gamma'-\gamma)(I-\gamma P^\pi)^{-1}(P^\pi)^T\right)^K d^\pi_{\gamma'}\end{aligned}$$

The norm of the residual term could be bounded as

$$\left\|(1-\gamma')\left((\gamma'-\gamma)(I-\gamma P^\pi)^{-1}P^\pi\right)^K d^\pi_{\gamma'}\right\|_\infty\leq(1-\gamma')\left(\frac{\gamma'-\gamma}{1-\gamma}\right)^{K+1}\to 0.$$

□

With a similar motivation as expansions in the primal space, we define the K^{th} order expansion by truncating to first $K+1$ terms,

$$d^{\pi}_{x,K,\gamma,\gamma'} \coloneqq \frac{1-\gamma'}{1-\gamma}\sum_{k=0}^{K}((\gamma'-\gamma)(I-\gamma P^{\pi})^{-1}P^{\pi})^{k}d^{\pi}_{x,\gamma} \tag{24}$$

The following result formalizes the connection between the K^{th} order dual approximation to the visitation distribution $d^{\pi}_{K,\gamma,\gamma'}$ and the primal approximation to the value function at state x, $V^{\pi}_{K,\gamma,\gamma'}(x)$.

Proposition B.2. The K^{th} order primal and dual approximations are related by the following equality for any $K \geq 0$,

$$V^{\pi}_{K,\gamma,\gamma'}(x) = (1-\gamma')^{-1}\left(d^{\pi}_{x,K,\gamma,\gamma'}\right)^{T} r^{\pi} \tag{25}$$

Proof. The proof follows by expanding out the RHS of the equation. Recall the definition of $d^{\pi}_{K,\gamma,\gamma'}$,

$$\begin{aligned}
(d^{\pi}_{K,\gamma,\gamma'})^{T} &= \frac{1-\gamma'}{1-\gamma}\sum_{k=0}^{K}(d^{\pi}_{\gamma})^{T}\left((\gamma'-\gamma)\left(I-\gamma P^{\pi}\right)^{-1}\right)^{k} \\
&= (1-\gamma')\sum_{k=0}^{K}\delta_x^{T}(I-\gamma P^{\pi})^{-1}\left((\gamma'-\gamma)\left(I-\gamma P^{\pi}\right)^{-1}\right)^{k} \\
&= (1-\gamma')\delta_x^{T}\left[\sum_{k=0}^{K}\left((\gamma'-\gamma)(I-\gamma P^{\pi})^{-1}P^{\pi}\right)^{k}\right]\cdot(I-\gamma P^{\pi})^{-1}.
\end{aligned}$$

Now multiply the RHS by r^{π} and recall that $V^{\pi}_{\gamma} = (I-\gamma P^{\pi})^{-1}r^{\pi}$, we conclude the proof,

$$\text{RHS} = \frac{1-\gamma'}{1-\gamma}\delta_x^{T}\left[\sum_{k=0}^{K}\left((\gamma'-\gamma)(I-\gamma P^{\pi})^{-1}P^{\pi}\right)^{k}\right]V^{\pi}_{\gamma} = (1-\gamma')\delta_x^{T}V^{\pi}_{K,\gamma,\gamma'} = (1-\gamma')V^{\pi}_{K,\gamma,\gamma'}(x).$$

□

Proposition B.2 shows that indeed, the K^{th} order approximation of the value function is equivalent to the K^{th} order approximation of the visitation distribution in the dual space. It is instructive to consider the special case $K=1$.

C. Details on Taylor expansion Q-function advantage estimation

Proposition C.1. Let $Q^{\pi}_{\gamma} \in \mathbb{R}^{\mathcal{X}\times\mathcal{A}}$ be the vector advantage functions. Let $\overline{P}^{\pi} \in \mathbb{R}^{(\mathcal{X}\times\mathcal{A})\times(\mathcal{X}\times\mathcal{A})}$ be the transition matrix such that $\overline{P}^{\pi}(x,a,x',a') = \pi(x'|x')p(x'|x,a)$. Define the K^{th} order Taylor expansion of advantage as $Q^{\pi}_{K,\gamma,\gamma'} \coloneqq \sum_{k=0}^{K}((\gamma'-\gamma)(I-\gamma\overline{P}^{\pi})^{-1}\overline{P}^{\pi})^{k}Q^{\pi}_{\gamma}$. Then $\lim_{K\to\infty}Q^{\pi}_{K,\gamma,\gamma'} = Q^{\pi}_{\gamma'}$ for any $\gamma < \gamma' < 1$.

Algorithm 4 Estimating the K^{th} term of the expansion (Q-function)

Require: A trajectory $(x_t,a_t,r_t)_{t=0}^{\infty} \sim \pi$ and discount factors $\gamma < \gamma' < 1$

1. Compute advantage function estimates $\widehat{Q}^{\pi}_{\gamma}(x_t,a_t)$ for states on the trajectory. For example, $\widehat{Q}^{\pi}_{\gamma}(x_t,a_t) = \sum_{t'\geq t}\gamma^{t'-t}r_{t'}$. One could also apply other alternatives (e.g., (Schulman et al., 2015b)) which potentiall reduce the variance of $\widehat{Q}^{\pi}_{\gamma}(x_t,a_t)$.
2. Sample K random time $\tau_i, 1 \leq i \leq K$, all i.i.d. geometrically distributed $\tau_i \sim \text{Geometric}(1-\gamma)$.
3. Return $\frac{(\gamma'-\gamma)^{K}}{(1-\gamma)^{K}}\widehat{Q}^{\pi}_{\gamma}(x_\tau,a_\tau)$, where $\tau = \sum_{i=1}^{K}\tau_i$.

Proof. The proof follows closely that of Taylor expansion based approximation to value functions in Proposition 3.2. Importantly, notice that here we define $\overline{P}^{\pi}$, which differs from P^{π} used in the derivation of value functions. In particular,

$\overline{P}^\pi(x,a,y,b) = p(y|x,a)\pi(b|y)$ for any $x,y \in \mathcal{X}, a,b \in \mathcal{A}$. Let r be the vector reward function. The Bellmen equation for Q-function is

$$Q^\pi_{\gamma'} = r + \gamma' \overline{P}^\pi Q^\pi_{\gamma'}.$$

Inverting the equation and applying the Woodbury inversion identity,

$$Q^\pi_{\gamma'} = (I - \gamma'\overline{P}^\pi)^{-1} r = \sum_{k=0}^{\infty} \left((\gamma' - \gamma) \left(I - \gamma \overline{P}^\pi \right)^{-1} \overline{P}^\pi \right)^k Q^\pi_\gamma$$

The above equality holds for all $\gamma < \gamma' < 1$ due to similar convergence argument as in Proposition 3.2. Truncating the infinite series at step K, we arrive at the K^{th} order expansion $Q^\pi_{K,\gamma,\gamma'}$. By construction, $\lim_{K\to\infty} Q^\pi_{K,\gamma,\gamma'} = Q^\pi_{\gamma'}$. □

D. Details on Taylor expansion update weighting

Proposition D.1. The following is true for all $K \geq 0$,

$$\rho_{x,K,\gamma,\gamma'}(x') = \mathbb{I}[x' = x] + \mathbb{E}_\pi \left[\sum_{t=1}^{\infty} f(K,t,\gamma,\gamma') \mathbb{I}[x_t = x'] \,\middle|\, x_0 = x \right],$$

Equivalently, the K^{th} order Taylor expansion of $V^\pi_{\gamma'}(x)$ is

$$V^\pi_{K,\gamma,\gamma'}(x) = V_\gamma(x) + \mathbb{E}_\pi \left[\sum_{t=1}^{\infty} f(K,t,\gamma,\gamma') V^\pi_\gamma(x_t) \,\middle|\, x_0 = x \right], \tag{26}$$

where $f(K,t,\gamma,\gamma') = \sum_{u=1}^{\min(K,t)} (\gamma' - \gamma)^u \gamma^{t-u} \binom{t-1}{t-u}$ is a weight function.

Proof. We start with a few lemmas.

Lemma D.2. For any $n \geq 0, k \geq 1$, define a set of k-dimensional vector $\left\{ x_1, ...x_k | x_i \in \mathbb{Z}_{\geq 0}, \sum_{i=1}^k x_i = n \right\}$ and let $F(n,k)$ be the size of this set. Then

$$F(n,k) = \binom{n+k-1}{k-1}.$$

Proof. By construction, the above set can be decomposed into smaller sets by fixing the value of x_k, i.e.,

$$\left\{ x_1, ...x_k | x_i \in \mathbb{Z}_{\geq 0}, \sum_{i=1}^k x_i = n \right\} = \cup_{s=0}^n \left\{ x_1, ...x_{k-1}, x_k | x_i \in \mathbb{Z}_{\geq 0}, \sum_{i=1}^k x_i = n, x_k = s \right\}$$

Since these sets do not overlap, we have a recursive formula, $F(n,k) = \sum_{s=0}^n F(n-s, k-1)$. Starting from the base case $F(n,1) = 1, \forall n \geq 0$, it is straightforward to prove by induction that for all $n \geq 0, k \geq 1$

$$F(n,k) = \binom{n+k-1}{k-1}.$$

□

Lemma D.3. Consider $V^\pi_{K+1,\gamma,\gamma'} - V^\pi_{K,\gamma,\gamma'}$ for $K \geq 0$. It can be shown that

$$V^\pi_{K+1,\gamma,\gamma'} - V^\pi_{K,\gamma,\gamma'} = (\gamma' - \gamma)^{K+1} \left(\sum_{t=0}^{\infty} F(t, K+1) \left(P^\pi\right)^t \right) \left(P^\pi\right)^{K+1} V^\pi_\gamma.$$

Proof. Starting with the definition,

$$\begin{aligned} V^\pi_{K+1,\gamma,\gamma'} - V^\pi_{K,\gamma,\gamma'} &= \left((\gamma'-\gamma)\left(I-\gamma P^\pi\right)^{-1} P^\pi\right)^{K+1} V^\pi_\gamma \\ &= \left((\gamma'-\gamma)\left(I-\gamma P^\pi\right)^{-1}\right)^{K+1} \left(P^\pi\right)^{K+1} V^\pi_\gamma, \end{aligned}$$

where for the second equality we use the fact that P^π commutes with $(I-\gamma P^\pi)^{-1}$. Then consider $\left((I-\gamma P^\pi)^{-1}\right)^{K+1}$,

$$\left((I-\gamma P^\pi)^{-1}\right)^{K+1} = \left(\sum_{t=0}^\infty \left(\gamma P^\pi\right)^t\right)^{K+1} = \sum_{s_1\geq 0} \ldots \sum_{s_{K+1}\geq 0} \left(\gamma P^\pi\right)^{\sum_{i=1}^{K+1} s_i} = \sum_{s=0}^\infty F(s,K+1)\left(\gamma P^\pi\right)^s.$$

Note that the last equality corresponds to a regrouping of terms in the infinite summation – instead of summing over $s_1, \ldots s_{K+1}$ sequentially, we count the number of examples such that $\sum_{i=1}^{K+1} s_i = s$ and then sum over s. This count is exactly $F(s, K+1)$ as defined in Lemma D.2. Hence the proof is completed. □

With the above lemmas, we are ready to prove the final result. We start by summing up all the differences of expansions,

$$\begin{aligned} V^\pi_{K,\gamma,\gamma'} &= V^\pi_{0,\gamma,\gamma'} + \sum_{k=0}^{K-1} \left(V^\pi_{k+1,\gamma,\gamma'} - V^\pi_{k,\gamma,\gamma'}\right) \\ &= V^\pi_\gamma + \sum_{k=0}^{K-1} (\gamma'-\gamma)^{k+1} \left(\sum_{t=0}^\infty F(t,k+1)\left(\gamma P^\pi\right)^t\right) \left(P^\pi\right)^{k+1} V^\pi_\gamma \\ &= V^\pi_\gamma + \sum_{t=0}^\infty \sum_{k=0}^{K-1} (\gamma'-\gamma)^{k+1} \gamma^{-k-1} F(t,k+1)\left(\gamma P^\pi\right)^{t+k+1} V^\pi_\gamma \\ &= V^\pi_\gamma + \sum_{t=0}^\infty \sum_{u=1}^{K} (\gamma'-\gamma)^{u} \gamma^{-u} F(t,u)\left(\gamma P^\pi\right)^{t+u} V^\pi_\gamma \\ &= V^\pi_\gamma + \sum_{s=0}^\infty \sum_{u=1}^{K} (\gamma'-\gamma)^{u} \gamma^{-u} F(s-u,u)\left(\gamma P^\pi\right)^{s} V^\pi_\gamma \\ &= V^\pi_\gamma + \sum_{s=1}^\infty \sum_{u=1}^{K} (\gamma'-\gamma)^{u} \gamma^{-u} F(s-u,u)\left(\gamma P^\pi\right)^{s} V^\pi_\gamma \\ &= V^\pi_\gamma + \sum_{s=1}^\infty \sum_{u=1}^{K} (\gamma'-\gamma)^{u} \gamma^{s-u} \binom{s-1}{u-1} \left(P^\pi\right)^{s} V^\pi_\gamma \\ &= V^\pi_\gamma + \sum_{s=1}^\infty \sum_{u=1}^{\min(K,s)} (\gamma'-\gamma)^{u} \gamma^{s-u} \binom{s-1}{u-1} \left(P^\pi\right)^{s} V^\pi_\gamma \end{aligned}$$

In the above derivation, we have applied the transformation $u = k+1, s = t+u$. Then we have modified the bound of the summation with the definition of $F(s-u,u)$ (in particular, if $s<u$, $F(s-u,u)=0$). If we index the x-th component of the vector, we recover the desired result.

□

D.1. Further discussions on the objectives

Recall that the full gradient $\nabla_\theta V^{\pi_\theta}_{\gamma'}(x)$ is

$$\nabla_\theta V^{\pi_\theta}_{\gamma'}(x) = \mathbb{E}_{x'\sim\rho^{\pi_\theta}_{\gamma,\gamma'}(\cdot;x)}\left[\nabla_\theta V^{\pi_\theta}_{\gamma}(x')\right] + \underbrace{\mathbb{E}_{x'\sim\rho^{\pi_\theta}_{\gamma,\gamma'}(\cdot;x)}\left[V^{\pi_\theta}_{\gamma}(x')\nabla_\theta \log \rho^{\pi_\theta}_{\gamma,\gamma'}(x';x).\right]}_{\text{second term}}$$

Consider the second term. Now, we derive this term in an alternative way which imparts more intuitions on why its estimation is challenging. Note that

$$V_{\gamma'}^{\pi_\theta}(x) = V_{\gamma}^{\pi_\theta}(x) + (\gamma' - \gamma)\mathbb{E}_\pi\left[\sum_{t=1}^{\infty}(\gamma')^{t-1}V_{\gamma}^{\pi_\theta}(x_t)\right]$$

The second term of the full gradient is equivalent to differentiating through the above expression, while keeping all $V_{\gamma}^{\pi_\theta}(x_t)$ fixed. This leads the following gradient

$$\text{second term} = (\gamma' - \gamma)(\gamma')^{-1}\mathbb{E}_\pi\left[\sum_{t=1}^{\infty}(\gamma')^{t}W_{\gamma,\gamma'}^{\pi_\theta}(x_t)\nabla_\theta \log \pi_\theta(a_t|x_t)\right].$$

Here, we introduce $W_{\gamma,\gamma'}^{\pi_\theta}(x_t) = \mathbb{E}_\pi\left[\sum_{s=0}^{\infty}(\gamma')^{s}V_{\gamma}^{\pi_\theta}(x_{t+s})\right]$, which is equivalent to a value function that treats $V_{\gamma}^{\pi_\theta}(x)$ as rewards and with discount factor γ'. Naturally, constructing an unbiased estimator of the second term of the full gradient requires estimating $W_{\gamma,\gamma'}^{\pi_\theta}$, which is difficult in at least two aspects: **(1)** in practice, value functions are already estimated, which could introduce additional bias and variance; **(2)** as a premise of our work, estimating discounted values with discount factor γ' is challenging potentially due to high variance.

E. Details on approximation errors with finite samples

Intuitively, as K increases, the K^{th} order expansion $V_{K,\gamma,\gamma'}^{\pi}$ approximates $V_{\gamma'}^{K}$ more accurately in expectation. However, in practice where all constituent terms of the approximation are built from the same batch of data, the variance might negatively impact the accuracy of the estimate.

To formalize such intuitions, we characterize the bias and variance trade-off under the phased TD-learning framework (Kearns and Singh, 2000). Consider estimating the value function $V_{\gamma}^{\pi}(x)$ under discount γ, with estimator $\widehat{V}_{\gamma}^{\pi}(x)$. At each iteration t, let $\Delta_t^\gamma := \max_{x\in\mathcal{X}}|V_{\gamma}^{\pi}(x) - \widehat{V}_{\gamma}^{\pi}(x)|$ be the absolute error of value function estimates $\widehat{V}_{\gamma}^{\pi}$. Assume from each state x, there are independent n trajectories generated under π, (Kearns and Singh, 2000) shows that commonly used TD-learning methods (e.g. TD(λ)) have error bounds of the following form with probability $1-\delta$,

$$\Delta_t^\gamma \le A(\gamma,\delta) + B(\gamma)\Delta_{t-1}^\gamma. \tag{27}$$

Here, the factor $A(\gamma,\delta)$ is an error term which characterizes the errors arising from the finite sample size n. As $n\to\infty$, $A(\gamma,\delta)\to 0$; the constant $B(\gamma)$ is a contraction coefficient that shows how fast the error decays in expectation. See Appendix E for details.

With the calculations of estimators $\widehat{V}_{\gamma}^{\pi}(x)$ as a subroutine, we construct the n-sample K^{th} order estimator $\widehat{V}_{K,\gamma,\gamma'}^{\pi}(x)$,

$$\widehat{V}_{K,\gamma}(x_0) = \sum_{k=0}^{K}\frac{1}{n}\sum_{i=1}^{n}(\gamma'-\gamma)^k\widehat{V}_{\gamma}^{\pi}(x_{i,k}), \tag{28}$$

where $x_{i,k}$ is sampled from $(P^\pi\cdot d_\gamma^\mu)^k(\cdot;x)$. Note that if $K=0$, Eqn (28) reduces to $\frac{1}{n}\sum_{i=1}^{n}\widehat{V}_{\gamma}^{\pi}(x_0)$, the estimator analyzed by (Kearns and Singh, 2000). We are interested in the error $\Delta_{K,t}^\gamma := \max_{x\in\mathcal{X}}|V_{\gamma'}^{\pi}(x) - \widehat{V}_{K,\gamma,\gamma'}^{\pi}(x)|$, measured against the value function of discount γ'. The following summarizes how errors propagate across iterations,

Proposition E.1. Assume all samples $x_{i,k}$ are generated independently. Define a factor $\varepsilon := \frac{1-(\gamma'-\gamma)^{K+1}}{1-(\gamma'-\gamma)}$. Then with probability at least $1-2\delta$ if $K\ge 1$ and probability $1-\delta$ if $K=0$, the following holds[1],

$$\Delta_{K,t}^\gamma \le \underbrace{\varepsilon(A(\gamma,\delta)+U)}_{\text{finite sample error}} + \underbrace{E(\gamma,\gamma',K)}_{\text{expected gap error}} + \underbrace{\varepsilon B(\gamma)}_{\text{contraction coeff}}\ \Delta_t^\gamma, \tag{29}$$

where $U = \sqrt{2\log\frac{2(K+1)}{\delta}/n}$ for $K\ge 1$ and $U=0$ if $K=0$. The expected gap error $E(\gamma,\gamma',K) = \left(\frac{\gamma'-\gamma}{1-\gamma}\right)^{K+1}\frac{R_{\max}}{1-\gamma}$ is defined in Proposition 3.2.

[1]The error bounds could be further improved, e.g., by adapting the concentration bounds at different steps $1\le k\le K$. Note that its purpose is to illustrate the bias and variance trade-off induced by the Taylor expansion order K.

Proof. Recall the results from (Kearns and Singh, 2000): Let $\Delta_t^\gamma \coloneqq \max_{x\in\mathcal{X}} |V_\gamma^\pi(x) - \widehat{V}_\gamma^\pi(x)|$. Then with probability at least $1-\delta$, the following holds

$$\Delta_t^\gamma \le A(\gamma,\delta) + B(\gamma)\Delta_{t-1}^\gamma.$$

In the following, we condition all analysis on the event set that the above inequality holds. Now, using $\widehat{V}_\gamma^\pi(x)$ as a subroutine, define the estimator for the K^{th} Taylor expansion as in Eqn (28),

$$\widehat{V}_{K,\gamma}(x_0) = \sum_{k=0}^{K} \frac{1}{n} \sum_{i=1}^{n} (\gamma' - \gamma)^k \widehat{V}_\gamma^\pi(x_{i,k}).$$

Define the error $\Delta_{K,t}^\gamma \coloneqq \max_{x\in\mathcal{X}} |V_{\gamma'}^\pi(x) - \widehat{V}_{K,\gamma}^\pi(x)|$, which is measured against the value function $V_{\gamma'}^\pi(x)$ with a higher discount factor γ'. Consider for a given starting state x_0,

$$\begin{aligned}
|V_{\gamma'}^\pi(x_0) - \widehat{V}_{K,\gamma}^\pi(x_0)| &= V_{\gamma'}^\pi(x_0) - V_{K,\gamma}^\pi(x_0) + V_{K,\gamma}^\pi(x_0) - \widehat{V}_{K,\gamma}^\pi(x) \\
&\le |V_{\gamma'}^\pi(x_0) - V_{K,\gamma}^\pi(x_0)| + V_{K,\gamma}^\pi(x_0) - \widehat{V}_{K,\gamma}^\pi(x_0) \\
&\le E(\gamma,\gamma',K) + \underbrace{V_{K,\gamma}^\pi(x_0) - \mathbb{E}\left[\widehat{V}_{K,\gamma}^\pi(x_0)\right]}_{\text{second term}} + \underbrace{\mathbb{E}\left[\widehat{V}_{K,\gamma}^\pi(x_0)\right] - \widehat{V}_{K,\gamma}^\pi(x_0)}_{\text{third term}}.
\end{aligned}$$

Now, we bound each term in the equation above. Recall $\varepsilon \coloneqq \sum_{k=0}^{K}(\gamma'-\gamma)^k = \frac{1-(\gamma'-\gamma)^{K+1}}{1-\gamma'+\gamma}$. The second term is bounded as follows

$$V_{K,\gamma}^\pi(x_0) - \mathbb{E}\left[\widehat{V}_{K,\gamma}^\pi(x_0)\right] \le \varepsilon\Delta_t^\gamma.$$

The third term is bounded by applying concentration bounds. Recall that the estimator $\widehat{V}_{K,\gamma}^\pi(x_0) \coloneqq \sum_{k=0}^{K}\frac{1}{n}\sum_{i=1}^{n}(\gamma'-\gamma)^k\widehat{V}_\gamma^\pi(x_{i,k})$ decomposes into $K+1$ estimators, each being an average over n i.i.d. samples drawn from the K^{th} step visitation distribution $(P^\pi\cdot d_\gamma^\pi)^k, 0\le k\le K$. Applying similarly naive techniques in (Kearns and Singh, 2000), we bound each of the $K+1$ terms individually and then take a union bound over all $K+1$ terms. This implies that, with probability at least $1-\delta$, the following holds

$$\mathbb{E}\left[\widehat{V}_{K,\gamma}^\pi(x_0)\right] - \widehat{V}_{K,\gamma}^\pi(x_0) \le \varepsilon U = \varepsilon\sqrt{2\log\frac{2(K+1)}{\delta}/n}.$$

Aggregating all results, we have

$$\begin{aligned}
|V_{\gamma'}^\pi(x_0) - \widehat{V}_{K,\gamma}^\pi(x_0)| &\le E(\gamma,\gamma',K) + \varepsilon\Delta_t^\gamma + \varepsilon U \\
&\le \varepsilon(A(\gamma,\delta)+U) + E(\gamma,\gamma',K) + \varepsilon B(\gamma)\Delta_{t-1}^\gamma.
\end{aligned}$$

This holds with probability at least $(1-\delta)^2 \ge 1-2\delta$.

□

Bias-variance trade-off via K. The error terms come from two parts: the first term contains errors $A(\gamma,\delta)$ in the subroutine estimator $\widehat{V}_\gamma^\pi(x)$, and its propagated errors through the sampling of K^{th} order approximations for $1\le k\le K$ (shown via the multiplier ε). This first term also contains U, a concentration bound that scales with $O(\sqrt{\log K})$, which shows that the variance of the overall estimator grows with K. This first error term scales with $\sqrt{n}$ and vanishes as the number of samples increases. The second term is due to the gap between the expected K^{th} order Taylor expansion and $V_{\gamma'}^\pi(x_0)$, which decreases with K and does not depend on sample size n. The new contraction coefficient is $\varepsilon B(\gamma)$, where it can be shown that $\varepsilon \in [1, \frac{1}{1-\gamma'+\gamma}]$. Since typical estimators have $B(\gamma)\le\gamma$, in general $\varepsilon B(\gamma) < 1$ and the error contracts with respect to Δ_t. In general, the contraction becomes slower as K increases. For example, for TD(λ), $B(\gamma) = \frac{(1-\lambda)\gamma}{1-\gamma\lambda}$.

F. Further experiment details

Below, we provide further details on experiment setups along with additional results.

F.1. Further details on the toy example

We presented a toy example that highlighted the trade-off between bias and variance, mediated by the order parameter K. Here, we provide further details of the experiments.

Toy MDP. We consider tabular MDPs with $|\mathcal{X}| = 10$ states and $|\mathcal{A}| = 2$ actions. The transition table $p(y|x, a)$ is drawn from a Dirichlet distribution $(\alpha, \ldots, \alpha)$ for $\alpha = 0.01$. Here, α is chosen such that the MDP is not very communicative (i.e., the distribution $p(\cdot|x, a)$ concentrates only on a few states). The rewards are random $r(x, a) = \overline{r}(x, a)(1 + \varepsilon)$ where $\varepsilon \sim \mathcal{N}(0, 0.2^2)$ and mean rewards $\overline{r}(x, a)$ are drawn from Uniform$(0, 1)$ and fixed for the problem.

F.2. Deep RL algorithms

Proximal policy optimization (PPO). PPO (Schulman et al., 2017) implements a stochastic actor $\pi_\theta(a|x)$ as a Gaussian distribution $a \sim \mathcal{N}(\mu_\theta(x), \sigma^2\mathbb{I})$ with state-conditional mean $\mu_\theta(x)$ and a global standard deviation $\sigma^2\mathbb{I}$; and a value function $V_\phi(x)$. The behavior policy μ is the previous policy iterate. The policy is updated as $\widehat{A}^\mu_\gamma(x, a)\nabla_\theta \text{clip}(\frac{\pi_\theta(a|x)}{\mu(a|x)}, 1 - \varepsilon, 1 + \varepsilon)$ with $\varepsilon = 0.2$[2]. The advantages $\widehat{A}^\mu_\gamma(x, a)$ estimated using generalized advantage estimation (GAE, (Schulman et al., 2015b)) with $\gamma = 0.99, \lambda = 0.95$. Value functions are trained by minimizing $(V_\phi(x) - R(x))^2$ with returns $R(x) = V_{\phi'}(x) + \widehat{A}^\mu_\gamma(x, a)$ with ϕ' being a prior parameter. Both parameters θ, ϕ are trained with the Adam optimizer (Kingma and Ba, 2014) with learning rate $\alpha = 3 \cdot 10^{-4}$. We adopt other default hyper-parameters in (Dhariwal et al., 2017), for details, please refer to the code base.

Trust region policy optimization (TRPO). TRPO (Schulman et al., 2015b) implements the same actor-critic pipeline as PPO, the difference is in the updates. Instead of enforcing a *soft* clipping constraint, TRPO enforces a strict KL-divegergence constraint $\mathbb{E}_{x\sim\mu}\left[\text{KL}(\pi_\theta(\cdot|x), \mu(\cdot|x)\right] \leq \varepsilon$ with $\varepsilon = 0.01$. The policy gradient is computed as $\widehat{A}^\mu_\gamma(x, a)\nabla_\theta \log \pi_\theta(a|x)$, and then the final update is constructed by approximately solving a constrained optimization problem, see (Schulman et al., 2015a) for details. The scale of the final update is found through a line search, to ensure that the KL-divergence constraint is satisfied. The implementations are based in (Achiam and OpenAI, 2018).

F.3. Deep RL architecture

Across all algorithms, the policy $\pi_\theta(a|x) = \mathcal{N}(\mu_\theta(x), \sigma^2\mathbb{I})$ has a parameterized mean $\mu_\theta(x)$ and a single standard deviation σ^2. The mean $\mu_\theta(x)$ is a 2-layer neural network with hidden units $h = 64$, and $f(x) = \tanh(x)$ activation functions. The output layer does not have any activation functions; The value function $V_\phi(x)$ is a 2-layer neural network with hidden units $h = 64$ and $f(x) = \tanh(x)$ as activation functions. The output layer does not have any activation functions.

F.4. Additional deep RL experiment results

F.4.1. Taylor expansion Q-function estimation: ablation study on η

Recall that throughout the experiments, we choose $K = 1$ and construct the new Q-function estimator as a mixture of the default estimator and Taylor expansion Q-function estimator. In particular, the final Q-function estimator is

$$\widehat{Q}(x, a) = (1 - \eta)\widehat{Q}^\pi_\gamma(x, a) + \eta\widehat{Q}^\pi_{K,\gamma,\gamma'}(x, a).$$

We choose $\eta \in [0, 1]$ such that it balances the numerical scales of the two combining estimators. In our implementation, we find that the algorithm performs more stably when η is small in the absolute scale. In Figure 5(a)-(b), we show the ablation study on the effect of η, where we vary $\eta \in [0.01, 0.03]$. The y-axis shows the normalized performance against PPO baselines (which is equivalent to $\eta = 0$), such that the PPO baseline achieves a normalized performance of 1.

Overall, we see on different tasks, η impacts the performance differently. For example: on HalfCheetah(B), better performance is achieved with larger values of η, this is consistent with the observation that PPO with $\gamma = 0.999$ also achieves better performance; on Ant(B), however, as η increases from zero, the performance increases marginally before degrading. In Figure 5, we show the median and mean performance across all tasks. Note that in general, the average performance increases as η increases from zero, but later starts to decay a bit. When accounting for the effect of performance variance across all tasks, we chose $\eta = 0.01$ as the fixed hyper-parameter throughout experiments in the main paper.

[2]The exact PPO update is more complicated than this. Refer to (Schulman et al., 2017) for the exact formula.

Further details on computing $\widehat{Q}^{\pi}_{K,\gamma,\gamma'}(x,a)$. Below we assume $K=1$. In Algorithm 4, we showed we can construct unbiased estimates of $Q^{\pi}_{K,\gamma,\gamma'}(x,a)$ using $\widehat{Q}^{\pi}_{\gamma}(x,a)$ as building blocks. With a random time $\tau \sim \text{Geometric}(1-\gamma)$, the estimator takes the following form

$$\widehat{Q}_{K,\gamma,\gamma'}(x_t,a_t) = \widehat{Q}^{\pi}_{\gamma}(x_t,a_t) + \frac{\gamma'-\gamma}{1-\gamma} Q^{\pi}_{\gamma}(x_{t+\tau},a_{t+\tau}).$$

However, since the estimator is based on a single random time, it can have high variance. To reduce variance, we propose the following procedure: let (x_t,a_t) be the target state-action pair, we can compute the estimate as

$$\widehat{Q}_{K,\gamma,\gamma'}(x_t,a_t) = \widehat{Q}^{\pi}_{\gamma}(x_t,a_t) + \frac{\gamma'-\gamma}{1-\gamma} \sum_{s=1}^{H} \frac{\gamma^s}{\sum_{s'=1}^{H}\gamma^{s'}} Q^{\pi}_{\gamma}(x_{t+s},a_{t+s}).$$

When $H=\infty$, the above estimator corresponds to an estimator which marginalizes over the random time. This should achieve variance reduction compared to the random time based estimate in Algorithm 4. However, then the estimate requires computing cumulative sums over an infinite horizon (or in general a horizon of T), which might be computationally expensive. To mitigate this, we propose to truncate the above summation up to $H=10$ steps. This choice of H aims to achieve a trade-off between computation efficiency and variance. Note that this estimator was previously introduced in (Tang et al., 2020) for off-policy learning.

F.4.2. Taylor expansion update weighting: ablation on K

In Figure 5(c)-(d), we carry out ablation study on the effect of K for the update weighting. Recall that K interpolates two extremes: when $K=0$, it recovers the vanilla PG (Sutton et al., 2000) while when $K=\infty$, it recovers the deep RL heuristic update. We expect an intermediate value of K to achieve some trade-off between bias and variance of the overall update.

In Figure 5(c), we see the effect on individual environments. The effect is case dependent. For HalfCheetah(G), larger K improves the performance; however, for Walker(G), the improvement is less prominent over a large range of K. When aggregating the performance metric in Figure 5(d), we see that intermediate values of K indeed peak in performance. We see that on average, both $K=10$ and $K=100$ achieve locally optimal mean performance, while $K=10$ also achieves the locally optimal median performance.

Note on how the practical updates impact the effect of K. Based on our theoretical analysis, when $K=0$ the update should recover the vanilla PG (Sutton et al., 2000), which is generally considered too conservative for the undiscounted objective in Eqn (1). However, in practice, as shown in Figure 5(d), the algorithm does not severely underperform even when $K=0$. We speculate that this is because practical implementations of PG updates use batches of data instead of the full trajectories. This means that the relative weights $w(t)$ of the local gradients $\widehat{Q}_t \nabla_\theta \log \pi_\theta(a_t|x_t)$ are effectively self-normalized: $\widetilde{w}(t) \leftarrow \frac{w(t)}{\sum w(t')}$ where the summation is over the time steps in a sampled mini-batch. The self-normalized weights $\widetilde{w}(t)$ are increased in the absolute scale relative to $w(t)$ and partly offset the effect of an initially aggressive discount $w(t)=\gamma^t$.

F.4.3. Comparison to results in (Romoff et al., 2019)

Recently, Romoff et al. (2019) derived a recursive relations between differences value functions defined with different discount factors. This was shown in Lemma 4.1. Given a sequence of discount factors $\gamma_1 < \gamma_2 < \ldots < \gamma_N < \gamma'$, they derived a value function estimator to $V^{\pi}_{\gamma'}(x)$ based on recursive bootstraps of value function differences $V^{\pi}_{\gamma_i}(x) - V^{\pi}_{\gamma_{i-1}}(x')$. Because they aim at recovering the exact value functions, this estimator could be interpreted as similar to Taylor expansions but with $K=\infty$.

Different from their motives, we focus on the trade-off achieved by intermediate values of K. We argued that by using $K=0$, the estimate might be too conservative; however, using $K=\infty$ might be challenging due to the variance induced in the recursive bootstrapping procedure. Though it is not straightforward to theoretically show, we conjecture that using the Taylor expansion Q-function estimator with $K=\infty$ is as difficult as directly estimating $V^{\pi}_{\gamma'}(x)$.

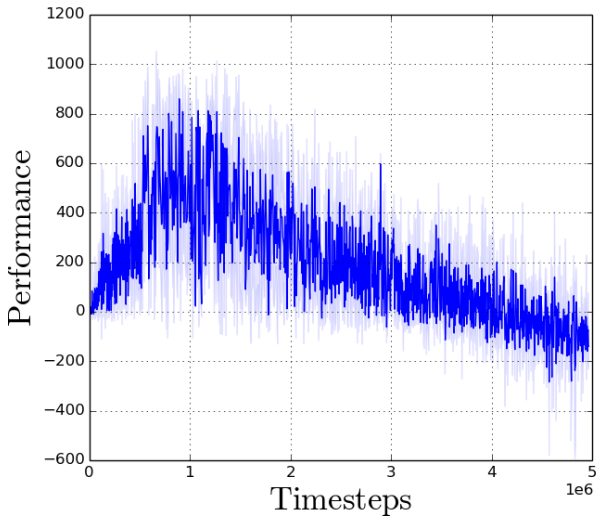
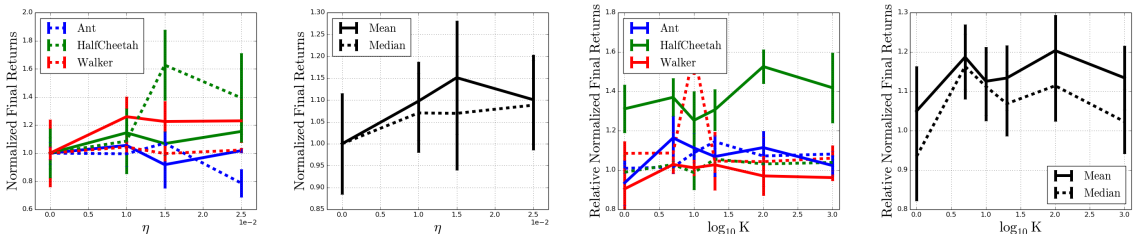


Figure 4. Learning curves generated by running the open source implementation of (Romoff et al., 2019) on Walker2d(G), averaged across 5 runs. There is little progress of learning for the algorithm on other benchmark tasks.



(a) Ablation on η (individual) (b) Ablation on η (average) (c) Ablation on K (individual) (d) Ablation on K (average)

Figure 5. Ablation study of hyper-parameters. We study two hyper-parameters: (a) η (b) K. In both cases, we calculate the task-dependent normalized final returns after training for 10^6 steps. See Appendix F for how such normalized returns are computed. In (a), normalized returns are computed with respect to $\eta = 0$ (i.e, the PPO baseline), such that when $\eta = 0$, the normalized returns are ones; in (b), normalized returns are computed with respect to the default PPO baseline, such that values of ones imply that the baseline performs the same as the default PPO baseline. Dashed curves (bullet tasks) and solid curves (gym tasks) are both mean scores averaged over 5 seeds.

Empirical comparison. The base algorithm of (Romoff et al., 2019) is PPO(Schulman et al., 2017). Their algorithm uses the recursive bootstraps to estimate Q-functions and advantage functions. The new estimate is used as a direct plug-in replacement to $\widehat{Q}^{\pi}_{\gamma}(x,a)$ and $\widehat{A}^{\pi}_{\gamma}(x,a)$ adopted in the PPO algorithm. We run experiments with the open source implementation of (Romoff et al., 2019) from the original authors[3]. We evaluate the algorithm's performance over continuous control benchmark tasks. We applied the default configurations from the code base with minimum changes to run on continuous problems (note that (Romoff et al., 2019) focused on a few discrete control problems). Overall, we find that the algorithm does not learn stably (see Figure 4).

G. Extensions of update weighting techniques to off-policy algorithms

Below, we show that techniques developed in this paper could be extended to off-policy learning algorithms. We provide both details in theoretical derivations, algorithms, as well as experimental results.

[3]See https://github.com/facebookresearch/td-delta.

G.1. Off-policy actor-critic algorithms

Off-policy actor-critics (Mnih et al., 2015; Lillicrap et al., 2015) maintain a deterministic policy $\pi_\theta(x)$ and a Q-function critic $Q_\phi(x,a)$. The agent takes exploratory actions under the environment, and saves data (x_t, a_t, r_t) into a common replay buffer $\mathcal{D}$. At training time, the algorithm samples data from the replay to update parameters. The policy is updated via the deterministic policy gradient (Silver et al., 2014), $\theta \leftarrow \theta + \alpha \nabla_\theta \mathbb{E}_\mu \left[Q_\phi(x, \pi_\theta(x))\right]$, where μ is implicitly defined by the past behavior policy.

Deep deterministic policy gradient (DDPG). DDPG (Lillicrap et al., 2015) maintains a deterministic policy network $\pi_\theta(a|x) \equiv \pi_\theta(x)$ and a Q-function critic $Q_\phi(x,a)$. The algorithm explores by executing a perturbed policy $a = \varepsilon + \pi_\theta(x)$ where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ for $\sigma = 0.1$, and then saves the data (x, a, r, x') into a replay buffer $\mathcal{D}$. At training time, the behavior data is sampled uniformly from the replay buffer $(x_i, a_i, r_i, x_i')_{i=0}^{B-1} \sim \mathcal{U}(\mathcal{D})$ with $B = 100$. The critic is updated via TD(0), by minimizing: $\frac{1}{B}\sum_{i=0}^{B-1}(Q_\phi(x_i,a_i) - Q_{\text{target}}(x_i,a_i))^2$ where $Q_{\text{target}}(x_i,a_i) = r_i + \gamma Q_{\phi'}(x_i', \pi_{\theta'}(x_i'))$, where θ', ϕ' are delayed versions of θ, ϕ respectively (Mnih et al., 2015). The policy is updated by maximizing $\frac{1}{B}\sum_{i=0}^{B-1} Q_\phi(x_i, \pi_\theta(x_i))$ with respect to θ. Both parameters θ, ϕ are trained with the Adam optimizer (Kingma and Ba, 2014) with learning rate $\alpha = 10^{-4}$. We adopt other default hyper-parameters in (Achiam and OpenAI, 2018), for details, please refer to the code base.

Twin-delayed DDPG (TD3). TD3 (Fujimoto et al., 2018) adopts the same training pipeline and architectures as DDPG. TD3 also adopts two critic networks $Q_{\phi_1}(x,a), Q_{\phi_2}(x,a)$ with parameters ϕ_1, ϕ_2, in order to minimize the over-estimation bias (Hasselt, 2010; Van Hasselt et al., 2016).

Soft actor-critic. SAC (Haarnoja et al., 2018) adopts a similar training pipeline and architectures as TD3. A major conceptual difference is that SAC is based on the maximum-entropy formulation of RL (Ziebart et al., 2008; Fox et al., 2016). The Q-function is augmented by entropy regularization bonus and the policy is optimized such that it does not collapse to a deterministic policy.

G.2. Architecture

All algorithms share the same architecture. The policy network $\pi_\theta(x)$ takes as input the state x, and is a 2-layer neural network with hidden units $h = 256$ and $f(x) = \text{relu}(x)$ activation functions. The output is squashed by $f(x) = \tanh(x)$ to comply with the action space boundaries; The critic $Q_\phi(x,a)$ takes a concatenated vector $[x,a]$ as inputs, is 2-layer neural network with hidden units $h = 256$ and $f(x) = \text{relu}(x)$ activation functions. The output does not have any activation functions.

For stochastic policies, the policy network parameterizes a Gaussian also parameterizes a log standard deviation vector $\log\sigma(x)$, which is a neural network with the same architecture above. The stochastic output is a reparameterized function $a = \pi_\theta(x) + \exp(\log\sigma(x)) \cdot \varepsilon$ where the noise $\varepsilon \sim \mathcal{N}(0,1)$. Finally, the action output is squashed by $\tanh(x)$ to comply with the action boundary (Haarnoja et al., 2018).

G.3. Algorithm details for update weighting

To derive an update based on update weighting, we start with the undiscounted on-policy objective $V_{\gamma'}(x) = \mathbb{E}_{x' \sim \rho^{\pi_\theta}_{x,\gamma,\gamma'}}\left[V_\gamma^{\pi_\theta}(x')\right]$. Given behavior data generated under μ, we abuse the notation and also use μ to denote the state distribution under μ (usually implicitly defined by sampling from a replay buffer $\mathcal{D}$). By rewriting the objective with importance sampling (IS),

$$V_{\gamma'}^{\pi_\theta}(x) = \mathbb{E}_{x' \sim \rho^{\pi_\theta}_{x,\gamma,\gamma'}}\left[V_\gamma^{\pi_\theta}(x')\right] = \mathbb{E}_{x' \sim \mu}\left[\frac{\rho^{\pi_\theta}_{x,\gamma,\gamma'}(x')}{\mu(x')} V_\gamma^{\pi_\theta}(x')\right], \tag{30}$$

we derive an off-policy learning objective. By dropping a certain terms (see (Degris et al., 2012) for details about the justifications for dropping such terms), we can derive the IS-based gradient update

$$\mathbb{E}_{x' \sim \mu}\left[\frac{\rho^{\pi_\theta}_{x,\gamma,\gamma'}(x')}{\mu(x')} \nabla_\theta V_\gamma^{\pi_\theta}(x')\right] \approx \mathbb{E}_{x' \sim \mu}\left[\frac{\rho^{\pi_\theta}_{x,\gamma,\gamma'}(x')}{\mu(x')} \nabla_\theta Q_\phi\left(x', \pi_\theta(x')\right)\right]$$

To render the update feasible, we need to estimate the ratio $\frac{\rho^{\pi_\theta}_{x,\gamma,\gamma'}(x')}{\mu(x')}$. Inspired by (Sinha et al., 2020), we propose to maintain a *fast replay buffer* $\mathcal{D}_f$ which contains the most recent sampled data (which implicitly defines $\rho^{\pi_\theta}_{x,\gamma,\gamma'}$), then the

estimator w_ψ is trained to estimate the density ratio between $\mathcal{D}$ (which implicitly defines μ) and $\mathcal{D}_f$. See Appendix F for further details. The full off-policy actor-critic algorithm is summarized in Algorithm G.3. In practice, we implement a undiscounted uniform distribution instead of $\rho^\pi_{x,\gamma,\gamma'}(x')$ with $\gamma' = 1$. The main motivation is that this distribution is much easier to specify as it corresponds to sampling from the replay buffer uniformly without discounts, as explained below.

As an important observation for practical implementations, note that

$$\rho^\pi_{x,\gamma,\gamma'}(x') = \frac{\gamma}{\gamma'}\mathbb{I}[x_0 = x'] + (\gamma' - \gamma)\mathbb{E}_\pi\left[\sum_{t\geq 1}(\gamma')^{t-1}\mathbb{I}[x_t = x'] \,\middle|\, x_0 = x\right]$$

when setting $\gamma' = 1$, we see that the second term of the distribution is proportional to $\mathbb{E}_\pi\left[\sum_{t\geq 1}\mathbb{I}[x_t = x'] \,\middle|\, x_0 = x\right]$, which corresponds to a uniform distribution over states on sampled trajectories, *without* discounting. This will make implementations much simpler. We will see that this could also lead to performance gains. We leave Taylor expansion based extension of this method for future work.

Details on training the density estimator $w_\psi(x)$. The density estimator $w_\psi(x)$ is parameterized with exactly the same architecture as the policy network $\pi_\theta(x)$, except that its output activation is replaced by $\log(1 + \exp(x))$ to ensure that $w_\psi(x) > 0$. The off-policy actor-critic algorithm maintains an original buffer $\mathcal{D}$ of size $|\mathcal{D}| = 10^6$; in addition, we maintain a fast replay buffer $\mathcal{D}_f$ with $|\mathcal{D}_f| = 10^4$, which is used for saving the most recently generated data points. For ease of analysis, assume that the data sampled from $\mathcal{D}_f$ come from π_θ, while the data sampled from $\mathcal{D}$ come from μ.

To learn the ratio $\frac{\rho^{\pi_\theta}_{x,\gamma,\gamma'}(x')}{\mu(x')}$, we adopt a simple discriminative loss function as follows

$$L(\psi) = -\mathbb{E}_{x'\sim\rho^{\pi_\theta}_{x,\gamma,\gamma'}}\left[\log\frac{w_\psi(x')}{1+w_\psi(x')}\right] - \mathbb{E}_{x'\sim\mu}\left[\log\frac{1}{1+w_\psi(x')}\right] \approx -\mathbb{E}_{x\sim\mathcal{D}_f}\left[\log\frac{w_\psi(x')}{1+w_\psi(x')}\right] - \mathbb{E}_{x\sim\mathcal{D}}\left[\log\frac{1}{1+w_\psi(x')}\right].$$

The optimal solution to $\psi^* = \arg\min_\psi L(\psi)$ is $w_{\psi^*}(x') = \frac{\rho^{\pi_\theta}_{x,\gamma,\gamma'}(x')}{\mu(x')}$(assuming enough expressiveness). Then, the density estimator is used for weighting the policy update: when sampling a batch of B data from the buffer, the weight $w_\psi(x_i), 1 \leq i \leq B$ is computed for each data point x_i. Then the weights are normalized across batch $\widetilde{w}_i = \frac{w_\psi(x_i)^\tau}{\sum_{j=1}^B w_\psi(x_j)^\tau}$ where the inverse temperature is $\tau = 0.1$. Then $\widetilde{w}_i$ is used for weighting the such that the policy is updated as $\theta \leftarrow \theta + \alpha\frac{1}{B}\sum_{i=1}^B \widetilde{w}_i\nabla_\theta Q_\phi(x_i, \pi_\theta(x_i))$.

Algorithm 5 Update weighting Off-policy actor-critic

Require: policy $\pi_\theta(x)$, Q-function critic $Q_\phi(x, a)$, density estimator $w_\psi(x)$ and learning rate $\alpha \geq 0$

while not converged **do**

1. Collect data $(x_t, a_t, r_t) \sim \mu$ and save to the buffer $\mathcal{D}$ and the fast buffer $\mathcal{D}_f$
2. Estimate the density by the discriminative loss between $\mathcal{D}, \mathcal{D}_f$, such that $w_\psi(x') \approx \rho^{\pi_\theta}_{x,\gamma,\gamma'}(x')/\mu(x')$, where x is the initial state of the MDP.
3. Sample data from $(x_i, a_i, r_i)_{i=1}^B \sim \mathcal{D}$.

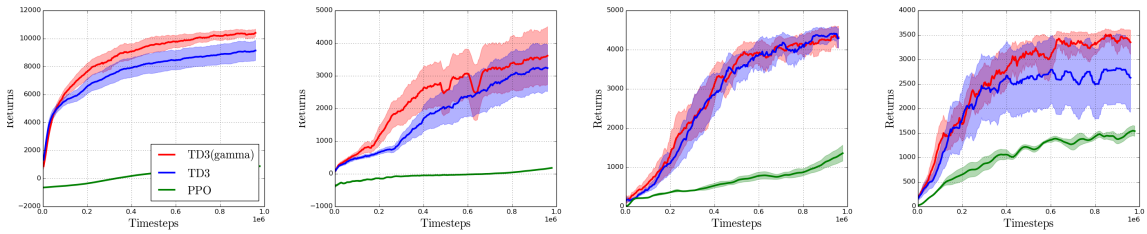
3(a). Update the Q-function critic $Q_\phi(x, a)$ via TD-learning, such that $Q_\phi(x, a) \approx Q^{\pi_\theta}_\gamma(x, a)$.

3(b). Update the policy parameter with the gradient $\theta \leftarrow \theta + \alpha\sum_{i=1}^B w_\psi(x_i)\nabla_\theta Q_\phi(x_i, \pi_\theta(x_i))$.

end while

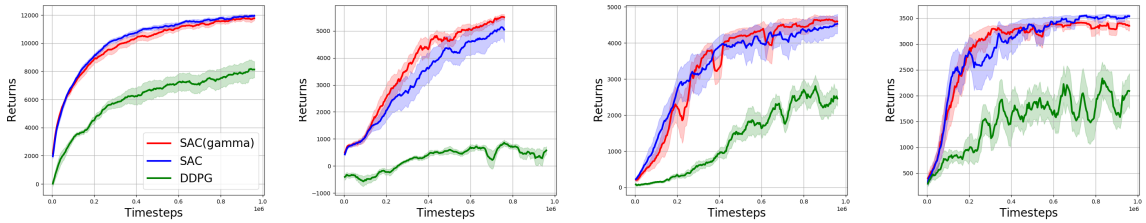
We carry out the update in Algorithm 2, where the density estimator $w_\psi(x)$ is trained based on a discriminative loss between $\mathcal{D}$ and $\mathcal{D}_f$. For any given batch of data $\{x_i\}_{i=1}^B$, we normalize the prediction $\widetilde{w}_i = w_\psi(x_i)^\tau / \sum_{j=1}^B w_\psi(x_j)^\tau$ with hyper-parameter $\tau = 0.1$ as similarly implemented in (Sinha et al., 2020). The temperature annealing moves $\widetilde{w}_i$ closer to a uniform distribution and tends to stabilize the algorithm. See Appendix F for further details.

Discussion on relations to other algorithms. Previous work focuses on re-weighting transitions to stabilize the training of critics. For example, prioritized replay (Schaul et al., 2015) prioritizes samples with high Bellman errors. Instead, Algorithm 2 reweighs samples to speed up the training of the policy. Our observation above also implies that when sampling from $\mathcal{D}, \mathcal{D}_f$ for training the estimates $w_\psi \approx \frac{\rho^{\pi_\theta}_{x,\gamma,\gamma'}(x')}{\mu(x')}$, it is not necessary to discount the transitions. This is in clear contrast



(a) HalfCheetah(G) (b) Ant(G) (c) Walker2d(G) (d) Hopper(G)

Figure 6. Evaluation of near off-policy actor-critic algorithms over continuous control domains. Each curve corresponds to a baseline algorithm averaged over 5 random seeds. TD3(γ) (red curve) consistently outperforms or performs similarly as other baselines.



(a) HalfCheetah(G) (b) Ant(G) (c) Walker2d(G) (d) Hopper(G)

Figure 7. Evaluation of near off-policy actor-critic algorithms over continuous control domains. Each curve corresponds to a baseline algorithm averaged over 5 random seeds. SAC(γ) (red curve) consistently outperforms or performs similarly as other baselines.

to prior work, such as (Sinha et al., 2020), where they propose to train $w_\psi(x') \approx d^{\pi_\theta}_{x,\gamma}(x')/d^{\mu}_{x,\gamma}(x')$, which is the fully discounted visitation distribution under γ based on the derivation of optimizing a discounted objective $V^{\pi_\theta}_\gamma(x)$.

Results. We build the algorithmic improvements based on TD3 (Fujimoto et al., 2018) and SAC (Haarnoja et al., 2018), and name the correspinding algorithms TD3(γ) and SAC(γ) respectively. We compare with TD3, SAC, and DDPG (Lillicrap et al., 2015), all of which are off-policy algorithms.

We first compare TD3(γ) with TD3 in Figure 6. To highlight the default sample efficiency of off-policy methods, we include PPO as a baseline as well. Across all four presented tasks, we see that TD(γ) performs similarly or marginally outperforms the TD3 baseline. To make concrete the comparison between final performance, we report the final score mean $\pm$ 0.5std of each algorithm in Table 1. As a default baseline, we also show the results of DDPG reported in (Achiam and OpenAI, 2018). Overall, TD3(γ) provides a modest yet consistent boost over baseline TD3.

Then we compare SAC(γ) with SAC in Figure 7 and Table 1. We see that SAC(γ) provides marginal performance gains over Walker2d and Ant, while it is slightly overperformed by baseline SAC for HalfCheetah and Hopper. We speculate that this is partly because the hyper-parameters of baseline SAC are well tuned on HalfCheetah, and it is difficult to achieve further significant gains without exhaustive hyper-parameter search. Overall, SAC(γ) is competitive compared to SAC.

Tasks	TD3(γ)	TD3	DDPG-v1
ANT(G)	$\mathbf{3601 \pm 879}$	$\mathbf{3269 \pm 686}$	≈ 1000
HALFCHEETAH(G)	$\mathbf{10350 \pm 279}$	9156 ± 718	≈ 8500
WALKER2D(G)	$\mathbf{4090 \pm 440}$	$\mathbf{4233 \pm 314}$	≈ 2000
HOPPER(G)	$\mathbf{3340 \pm 262}$	2626 ± 677	≈ 1800
Tasks	**SAC(γ)**	**SAC**	**DDPG-v2**
ANT(G)	$\mathbf{5572 \pm 115}$	4886 ± 530	706 ± 123
HALFCHEETAH(G)	11774 ± 96	$\mathbf{12059 \pm 91}$	7957 ± 527
WALKER2D(G)	$\mathbf{4626 \pm 165}$	$\mathbf{4522 \pm 269}$	2261 ± 147
HOPPER(G)	3384 ± 81	$\mathbf{3557 \pm 20}$	2024 ± 297

Table 1. Final performance of baseline algorithms over benchmark tasks. The final performance is computed as the mean scores over the last 10 iterations of each algorithm, averaged over 5 seeds. When compared with TD3, the performance of DDPG-v1 is taken from (Achiam and OpenAI, 2018); when compared with SAC, the performance is based on re-runs of the DDPG-v2 baselines with (Achiam and OpenAI, 2018). For each task, the best algorithms are highlighted in bold fonts (potentially with ties).