AVERAGE REWARD REINFORCEMENT LEARNING WITH MONOTONIC POLICY IMPROVEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

In continuing control tasks, an agent's average reward per time step is a more natural performance measure compared to the commonly used discounting framework since it can better capture an agent's long-term behavior. We derive a novel lower bound on the difference of the long-term average reward for two policies. The lower bound depends on the average divergence between the policies and on the so-called Kemeny constant, which measures to what degree the unichain Markov chains associated with the policies are well-connected. We also show that previous work based on the discounted return (Schulman et al., 2015; Achiam et al., 2017) results in a non-meaningful lower bound in the average reward setting. Based on our lower bound, we develop an iterative procedure which produces a sequence of monotonically improved policies for the average reward criterion. When combined with Deep Reinforcement Learning (DRL) methods, the procedure leads to scalable and efficient algorithms for maximizing the agent's average reward performance. Empirically we demonstrate the effectiveness of our method on continuing control tasks and show how discounting can lead to unsatisfactory performance.

1 INTRODUCTION

The goal of Reinforcement Learning (RL) is to build agents that can learn high-performing behaviors through trial-and-error interactions with the environment. Broadly speaking, modern RL tackles two kinds of problems: *episodic tasks* and *continuing tasks*. In episodic tasks, the agent-environment interaction can be broken into separate distinct episodes, and the performance of the agent is simply the sum of the rewards accrued within an episode. Examples of episodic tasks include training an agent to learn to play Go (Silver et al., 2016; 2018) or Atari video games (Mnih et al., 2013), where the episode terminates when the game ends. In continuing tasks, such as controlling robots with long operating lifespans (Peters & Schaal, 2008; Schulman et al., 2015; Haarnoja et al., 2018), there is no natural separation of episodes and the agent-environment interaction continues indefinitely. The performance of an agent in a continuing task is more difficult to quantify since even for bounded reward functions, the total sum of rewards is typically infinite.

One way of making the long-term reward objective meaningful for continuing tasks is to apply *discounting*, i.e., we maximize the discounted sum of rewards $r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots$ for some discount factor $\gamma \in (0, 1)$. This is guaranteed to be finite for any bounded reward function. However the discounted objective biases the optimal policy to choose actions that lead to high near-term performance rather than to high long-term performance. Such an objective — while useful in certain applications — is not appropriate when the goal is optimize long-term behavior. As argued in Chapter 10 of Sutton & Barto (2018) and in Naik et al. (2019), a more natural objective is to use the average reward received by an agent over every time-step. While the average reward setting has been extensively studied in the classical Markov Decision Process literature (Howard, 1960; Blackwell, 1962; Veinott, 1966; Bertsekas et al., 1995), it is much less commonly used in reinforcement learning. An important open question is whether recent advances in RL for the discounted reward criterion can be naturally generalized to the average reward setting.

One major source of difficulty with modern DRL algorithms lies in controlling the step-size for policy updates. In order to have better control over step-sizes, Schulman et al. (2015) constructed a lower bound on the difference between the expected discounted return for two arbitrary policies π and π'. The bound is a function of the divergence between these two policies and the discount factor.

Schulman et al. (2015) showed that iteratively maximizing this lower bound generates a sequence of monotonically improved policies in terms of their discounted return.

In this paper, we first show that the policy improvement theorem from Schulman et al. (2015) results in a non-meaningful bound in the average reward case. We then derive a novel result which lower bounds the difference of the average rewards based on the divergence of the policies. The bound depends on the average divergence between the policies and on the so-called Kemeny constant, which measures to what degree the unichain Markov chains associated with the policies are well-connected. We show that iteratively maximizing this lower bound guarantees monotonic average reward policy improvement. Similar to the discounted case, the problem of maximizing the lower bound can be approximated with DRL algorithms which can be optimized using samples collected in the environment. We describe in detail two such algorithms: Average Reward TRPO (ATRPO) and Average Cost CPO (ACPO), which are average reward versions of algorithms based on the discounted criterion (Schulman et al., 2015; Achiam et al., 2017). Using the MuJoCo simulated robotic benchmark, we carry out extensive experiments with the ATRPO algorithm and show that it is more effective than their discounted counterparts for these continuing control tasks. To our knowledge, this is one of the first paper to address DRL using the long-term average reward criterion.

2 PRELIMINARIES

Consider a Markov Decision Process (MDP) (Sutton & Barto, 2018) $(\mathcal{S}, \mathcal{A}, P, r, \mu)$ where the state space $\mathcal{S}$ and action space $\mathcal{A}$ are assumed to be finite. The transition probability is denoted by $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, the bounded reward function $r : \mathcal{S} \times \mathcal{A} \to [r_{\min}, r_{\max}]$, and $\mu : \mathcal{S} \to [0, 1]$ is the initial state distribution. Let $\pi = \{\pi(a|s) : s \in \mathcal{S}, a \in \mathcal{A}\}$ be a stationary policy, and Π is the set of all stationary policies. Here we discuss the two objective formulations for continuing control tasks: the average reward approach and discounted reward approach.

Average Reward Approach
In this paper, we will focus exclusively on *unichain* MDPs, which is when the Markov chain corresponding to every policy contains only one recurrent class and a finite but possibly empty set of transient states. The average reward objective is defined as:

$$\rho(\pi) := \lim_{N\to\infty} \frac{1}{N} \mathop{\mathbb{E}}_{\tau\sim\pi} \left[\sum_{t=0}^{N-1} r(s_t, a_t) \right] = \mathop{\mathbb{E}}_{\substack{s\sim d_\pi \\ a\sim\pi}} [r(s, a)]. \tag{1}$$

Here $d_\pi(s) := \lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-1} P(s_t = s|\pi) = \lim_{t\to\infty} P(s_t = s|\pi)$ is the *stationary state distribution under policy* π, $\tau = (s_0, a_0, \dots,)$ is a sample trajectory. We use $\tau \sim \pi$ to indicate that the trajectory is sampled from policy π, i.e. $s_0 \sim \mu$, $a_t \sim \pi(\cdot|s_t)$, and $s_{t+1} \sim P(\cdot|s_t, a_t)$. In the unichain case, the average reward $\rho(\pi)$ is state-independent for any policy π (Bertsekas et al., 1995). We express the *average-reward value function* as $V^\pi(s) := \mathbb{E}_{\tau\sim\pi} \left[\sum_{t=0}^{\infty} (r(s_t, a_t) - \rho(\pi)) \middle| s_0 = s \right]$ and *action-value function* as $Q^\pi(s, a) := \mathbb{E}_{\tau\sim\pi} \left[\sum_{t=0}^{\infty} (r(s_t, a_t) - \rho(\pi)) \middle| s_0 = s, a_0 = a \right]$. We define the *average reward advantage function* as $A^\pi(s, a) := Q^\pi(s, a) - V^\pi(s)$.

Discounted Reward Approach
For some discount factor $\gamma \in (0, 1)$, the discounted reward objective is defined as

$$\rho_\gamma(\pi) := \mathop{\mathbb{E}}_{\tau\sim\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right] = \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s\sim d_{\pi,\gamma} \\ a\sim\pi}} [r(s, a)]. \tag{2}$$

where $d_{\pi,\gamma}(s) := (1-\gamma) \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi)$ is known as the *future discounted state visitation distribution under policy* π. Note that unlike the average reward objective, *the discounted objective depends on the initial state distribution* μ. It can be easily shown that $d_{\pi,\gamma}(s) \to d_\pi(s)$ for all s as $\gamma \to 1$. The *discounted value function* is defined as $V^\pi_\gamma(s) := \mathbb{E}_{\tau\sim\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \middle| s_0 = s \right]$ and *discounted action-value function* $Q^\pi_\gamma(s, a) := \mathbb{E}_{\tau\sim\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \middle| s_0 = s, a_0 = a \right]$. Finally, the *discounted advantage function* is defined as $A^\pi_\gamma(s, a) := Q^\pi_\gamma(s, a) - V^\pi_\gamma(s)$.

It is well-known that $\lim_{\gamma \to 1}(1-\gamma)\rho_\gamma(\pi) = \rho(\pi)$, implying that the discounted and average reward objectives are equivalent in the limit as γ approaches 1 (Blackwell, 1962). We will further discuss the relationship between the discounted and average reward value functions in the supplementary materials and prove that $\lim_{\gamma \to 1} A^\pi_\gamma(s,a) = A^\pi(s,a)$ (see Corollary A.1).

3 Montonically Improvement Guarantees for Discounted RL

In many modern RL literature (Schulman et al., 2015; 2017; Abdolmaleki et al., 2018; Vuong et al., 2019), algorithms iteratively update policies within a local region, i.e., at iteration k we find policy π_{k+1} by maximizing $\rho_\gamma(\pi)$ within some region $D(\pi, \pi_k) \leq \delta$ for some divergence measure D. This approach allows us to control the step-size of each update using different choices of D and δ which can lead to better sample efficiency (Peters & Schaal, 2008). Schulman et al. (2015) derived a policy improvement bound based on a specific choice of D:

$$\rho_\gamma(\pi_{k+1}) - \rho_\gamma(\pi_k) \geq \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s\sim d_{\pi_k,\gamma} \\ a\sim\pi_{k+1}}} [A^{\pi_k}_\gamma(s,a)] - C \cdot \max_s [D_{\text{TV}}(\pi_{k+1} \parallel \pi_k)[s]] \tag{3}$$

where $D_{\text{TV}}(\pi' \parallel \pi)[s] := \frac{1}{2}\sum_a |\pi'(a|s) - \pi(a|s)|$ is the *total variation divergence* for policies π and π', and C is some constant which does not depend on the divergence term D_{TV}. Schulman et al. (2015) showed that by choosing π_{k+1} such that the right hand side of (3) is maximized, we are guaranteed to have $\rho_\gamma(\pi_{k+1}) \geq \rho_\gamma(\pi_k)$. This provided the theoretical foundation for an entire class of scalable policy optimization algorithms based on efficiently maximizing the right-hand-side of (3) (Schulman et al., 2015; 2017; Wu et al., 2017; Abdolmaleki et al., 2018; Vuong et al., 2019).

A natural question arises here is whether the iterative procedure described by Schulman et al. (2015) also guarantees improvement w.r.t. the average reward. Since the discounted and average reward objectives are equivalent when $\gamma \to 1$, one may assume that we can also lower bound the policy performance difference of the average reward objective by letting $\gamma \to 1$ for the bounds in Schulman et al. (2015). Unfortunately this results in a non-meaningful bound. We will demonstrate this through a similar policy improvement bound from Achiam et al. (2017) based on the average divergence but a similar argument can be made for the original bound from Schulman et al. (2015) (see supplementary material for proof and discussion).

Proposition 1. *Consider the following bound from Achiam et al. (2017)*

$$D^-_{\pi,\gamma}(\pi') \leq \rho_\gamma(\pi') - \rho_\gamma(\pi) \leq D^+_{\pi,\gamma}(\pi') \tag{4}$$

where

$$D^\pm_{\pi,\gamma}(\pi') = \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s\sim d_\pi \\ a\sim\pi}} \left[\frac{\pi'(a|s)}{\pi(a|s)} A^\pi_\gamma(s,a)\right] \pm \frac{2\gamma\epsilon_\gamma}{(1-\gamma)^2} \mathop{\mathbb{E}}_{s\sim d_\pi} [D_{TV}(\pi' \parallel \pi)[s]]$$

and $\epsilon_\gamma = \max_s \left|\mathbb{E}_{a\sim\pi'}[A^\pi_\gamma(s,a)]\right|$. *We have:*

$$\lim_{\gamma\to 1}(1-\gamma)D^\pm_{\pi,\gamma}(\pi') = \pm\infty \tag{5}$$

Since $\lim_{\gamma\to 1}(1-\gamma)(\rho_\gamma(\pi') - \rho_\gamma(\pi)) = \rho(\pi') - \rho(\pi)$, Proposition 1 says (4) becomes trivial when used on the average reward. This result is discouraging as it shows that the policy improvement guarantee from Schulman et al. (2015) does not appear to generalize to the average reward setting. In the next section, we will derive an alternative policy improvement bound for the average reward objective which can be used to generate monotonically improved policies w.r.t. the average reward.

4 Main Results

4.1 Average Reward Policy Improvement Theorem

Let $d_\pi \in \mathbb{R}^{|\mathcal{S}|}$ be the probability column vector whose components are $d_\pi(s)$, $P_\pi \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ be the transition matrix under policy π whose (s,s') component is $P_\pi(s'|s) = \sum_a P(s'|s,a)\pi(a|s)$, and $P^*_\pi = \lim_{t\to\infty} P^t_\pi$ be the limiting distribution for the transition matrix. We use $\|\cdot\|_p$ to denote the

operator norm of a matrix. In particular $\|\cdot\|_1$ and $\|\cdot\|_\infty$ are the maximum absolute column sum and maximum absolute row sum of a matrix respectively (Horn & Johnson, 2012).

Suppose we have a new policy π' obtained via some update rule from the current policy π. Similar to the discounted case, we would like to measure their performance difference $\rho(\pi') - \rho(\pi)$ using an expression which depends on π and some divergence metric between the two policies. The following identity shows that $\rho(\pi') - \rho(\pi)$ can be expressed using the advantange function of π.

Lemma 1. *For any two stochastic policies π and π':*

$$\rho(\pi') - \rho(\pi) = \lim_{N\to\infty} \frac{1}{N} \mathop{\mathbb{E}}_{\tau\sim\pi'} \left[\sum_{t=0}^{N-1} A^\pi(s_t, a_t) \right] = \mathop{\mathbb{E}}_{\substack{s\sim d_{\pi'} \\ a\sim\pi'}} [A^\pi(s, a)] \tag{6}$$

Lemma 1 is the an extension of the well-known policy difference lemma from Kakade & Langford (2002) to the average reward case. A similar result was proved by Neu et al. (2010) and Even-Dar et al. (2009). For completeness, We will provide a proof based on the Bellman equation as well as a simpler alternative proof in the supplementary material. Note that this expression depends on samples drawn from π'. However we can show through the following lemma that when d_π and $d_{\pi'}$ are "close," we can evaluate the expression in (6) using samples from d_π (see supplementary material for proof).

Lemma 2. *For any two stochastic policies π and π', the following bound holds:*

$$\mathop{\mathbb{E}}_{\substack{s\sim d_\pi \\ a\sim\pi'}} [A^\pi(s, a)] - 2\epsilon D_{TV}(d_{\pi'} \parallel d_\pi) \leq \rho(\pi') - \rho(\pi) \leq \mathop{\mathbb{E}}_{\substack{s\sim d_\pi \\ a\sim\pi'}} [A^\pi(s, a)] + 2\epsilon D_{TV}(d_{\pi'} \parallel d_\pi) \tag{7}$$

where $\epsilon = \max_s \left| \mathbb{E}_{a\sim\pi'(a|s)}[A^\pi(s, a)] \right|$.

Lemma 2 shows us how policy improvement is related to the stationary distribution underlying each policy. In order to study how policy improvement is connected to changes in the actual policies themselves, we need to analyze the relationship between changes in the policies and changes in stationary distributions. It turns out that the sensitivity of the stationary distributions in relation to the policies is related to the structure of the underlying Markov chain.

Let $M^\pi \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ be the *mean first passage time matrix* whose elements $M^\pi(s, s')$ is the expected number of steps it takes to reach state s' from s under policy π. The matrix $M^\pi(s, s')$ can be calculated via (Theorem 4.4.7 of Kemeny & Snell (1960))

$$M^\pi(s, s') = (I - Z^\pi + EZ^\pi_{\text{dg}})D^\pi \tag{8}$$

where $Z^\pi = (I - P_\pi + P^*_\pi)^{-1}$ is known as the *fundamental matrix of the Markov chain* (Kemeny & Snell, 1960), E is a square matrix consisting of all ones. The subscript 'dg' for some square matrix A refers to a diagonal matrix whose elements are the diagonals of A. $D^\pi \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ is a diagonal matrix whose elements are $1/d_\pi(s)$. One important property of mean first passage time is that given some policy π:

$$\kappa^\pi = \sum_{s'} d_\pi(s') M^\pi(s, s') \tag{9}$$

is a constant independent of the starting state s. This result is known as the *random target lemma* (Aldous & Fill, 1995). The constant κ^π is sometimes referred to as *Kemeny's constant* (Grinstead & Snell, 2012). This constant can be interpreted as the mean number of steps it takes to get to any goal state weighted by the steady-distribution of the goal states. This weighted mean does not depend on the starting state, as mentioned just above. The constant uses a single number to summarize how "well-connected" a Markov chain is. It can also be shown that $\kappa^\pi = \text{trace}(Z^\pi)$ (Grinstead & Snell, 2012). We then have the following result which connects the sensitivity of the stationary distribution to changes to the policy.

Lemma 3. *The divergence between the stationary distributions d_π and $d_{\pi'}$ can be upper bounded by the average divergence between policies π and π' as follows:*

$$D_{TV}(d_{\pi'} \parallel d_\pi) \leq (\kappa^{\pi'} - 1) \mathop{\mathbb{E}}_{s\sim d_\pi} [D_{TV}(\pi' \parallel \pi)[s]] \tag{10}$$

We wish to point out here that Achiam et al. (2017) showed a similar result to Lemma 3 in the discounted case where the change in $d_{\pi,\gamma}$ can be bounded in terms of the change in policy up to a multiplicative constant which only depends on the discount factor. In the discounted case, this is possible since a discounted MDP is like a finite-horizon MDP problem; in fact, it can be shown to be equivalent to a related finite horizon problem (Proposition 5.3.1, Puterman (1994)). The discount factor can be used to control the effective horizon where larger discount factors correspond to longer horizons. In fact, it can be easily shown that the multiplicative factor from Achiam et al. (2017) goes to infinity as $\gamma \to 1$, meaning that the bound is not useful for long horizon problems. In the average reward setting, the sensitivity of the stationary distribution with respect to the policy can vary depending on the chain structure and long-term behavior of the underlying Markov chain. This means that it is only natural that the multiplicative constant in Lemma 3 depends on the transition matrix.

This result is also highly intuitive, For very "well-connected" Markov chains where an agent can easily and quickly get to any state, this constant is relatively small and the stationary distributions are not sensitive to small changes in policy. On the other hand, for Markov chains that are "weakly connected," where on average, it can take a long time to get to some recurrent state in the state space, the factor can become very large. In this case small changes in the policy can have a large impact on the resulting stationary distributions.

The following theorem connects the average reward performance of two policies and their average divergence.

Theorem 1. *For any two stochastic policies π and π', the following bounds hold:*

$$\rho(\pi') - \rho(\pi) \leq \mathop{\mathbb{E}}_{\substack{s\sim d_\pi \\ a\sim\pi}} \left[\frac{\pi'(a|s)}{\pi(a|s)} A^\pi(s,a)\right] + 2\xi \mathop{\mathbb{E}}_{s\sim d_\pi} [D_{TV}(\pi' \parallel \pi)[s]] \tag{11}$$

$$\rho(\pi') - \rho(\pi) \geq \mathop{\mathbb{E}}_{\substack{s\sim d_\pi \\ a\sim\pi}} \left[\frac{\pi'(a|s)}{\pi(a|s)} A^\pi(s,a)\right] - 2\xi \mathop{\mathbb{E}}_{s\sim d_\pi} [D_{TV}(\pi' \parallel \pi)[s]] \tag{12}$$

where $\xi = (\kappa^{\pi'} - 1) \max_s \mathbb{E}_{a\sim\pi'} |A^\pi(s,a)|$.

Proof. Combine the bounds from Lemma 2 and Lemma 3. Then rewrite the expectation for $A^\pi(s,a)$ as an expectation w.r.t. π using importance sampling gives us the desired bound. □

The right-hand-side of the bounds in Theorem 1 are guaranteed to be finite. Similar to the discounted case, the multiplicative factor ξ provides a theoretical guidance on the step-sizes for policy updates (Schulman et al., 2015). The bound in Theorem 1 is given in terms of the TV divergence, however the KL divergence is more commonly used in practice. Vuong et al. (2019) compared various divergence measures and showed that the KL has superior empirical performance. The relationship between the TV divergence and KL divergence is given by Pinsker's inequality (Tsybakov, 2008), which says that for any two distributions p and q: $D_{\text{TV}}(p \parallel q) \leq \sqrt{D_{\text{KL}}(p\|q)/2}$. We can then show that

$$\mathop{\mathbb{E}}_{s\sim d_\pi} [D_{\text{TV}}(\pi' \parallel \pi)[s]] \leq \mathop{\mathbb{E}}_{s\sim d_\pi} [\sqrt{D_{\text{KL}}(\pi'\|\pi)[s]/2}] \leq \sqrt{\mathop{\mathbb{E}}_{s\sim d_\pi} [D_{\text{KL}}(\pi'\|\pi)][s]]/2} \tag{13}$$

where the second inequality comes from Jensen's inequality. The inequality in (13) shows that the bounds in Theorem 1 still hold when $\mathbb{E}_{s\sim d_\pi}[D_{\text{TV}}(\pi' \parallel \pi)[s]]$ is substituted with $\sqrt{\mathbb{E}_{s\sim d_\pi}[D_{\text{KL}}(\pi'\|\pi)][s]/2}$.

4.2 Approximate Policy Iteration

One direct consequence of Theorem 1 is that iteratively maximizing the right-hand-side of (12) generates a monotonically improving sequence of policies w.r.t. the average reward objective. Algorithm 1 gives an approximate policy iteration algorithm that produces such a sequence of policies.

Proposition 2. *Given an initial policy π_0, Algorithm 1 is guaranteed to generate a sequence of policies $\pi_1, \pi_2, \ldots$ such that $\rho(\pi_0) \leq \rho(\pi_1) \leq \rho(\pi_2) \leq \cdots$.*

Proof. At iteration k, $\mathbb{E}_{s\sim d_{\pi_k}, a\sim\pi}[A^{\pi_k}(s,a)] = 0$, $\mathbb{E}_{s\sim d_{\pi_k}}[D_{\text{KL}}(\pi\|\pi_k)[s]] = 0$ for $\pi = \pi_k$. By Equation (14) and Theorem 1, $\rho(\pi_{k+1}) - \rho(\pi_k) \geq 0$. □

Algorithm 1 Approximate Policy Iteration for Average Reward Objective

Initialize: π_0

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: Policy evaluation step: evaluate $A^{\pi_k}(s, a)$ for all s, a.
3: Policy improvement step:

$$\pi_{k+1} = \operatorname*{argmax}_{\pi} \left(\mathop{\mathbb{E}}_{\substack{s\sim d_{\pi_k} \\ a\sim\pi}} \left[A^{\pi_k}(s,a)\right] - \xi \sqrt{2 \mathop{\mathbb{E}}_{s\sim d_{\pi_k}} \left[D_{\mathrm{KL}}\left(\pi \| \pi_k\right)[s]\right]} \right) \tag{14}$$

where $\xi = (\kappa^{\pi} - 1) \max_s \mathbb{E}_{a\sim\pi} \left|A^{\pi_k}(s,a)\right|$

However, Algorithm 1 is difficult to implement in practice since it requires exact knowledge of the advantage function and transition matrix. Furthermore, calculating the term ξ is impractical for high dimensional problems. In the next section, we will introduce a sample-based algorithm which approximates the update rule given in Equation (14).

5 Practical Applications

As we have noted in the previous section, Algorithm 1 is not practical for problems with large state and action spaces and thus cannot be naïvely applied directly. In this section, we will discuss how Algorithm 1 and Theorem 1 can be used in practice to create algorithms which can effectively solve high dimensional DRL problems. In the Appendix C, we will also discuss how Theorem 1 can be used to solve DRL problems with safety constraints.

5.1 Average Reward Trust Region Policy Optimization

For DRL problems, it is common to consider some parameterized policy class $\Pi_\Theta \subseteq \Pi$. Our goal is to devise a computationally tractable version of Algorithm 1 for policies in Π_Θ, i.e., given a policy π_{θ_k} at iteration k, how do we obtain the best possible $\pi_{\theta_{k+1}}$? We can rewrite the unconstrained optimization problem in (14) as a constrained problem:

$$\underset{\pi_\theta \in \Pi_\theta}{\text{maximize}} \quad \mathop{\mathbb{E}}_{\substack{s\sim d_{\pi_{\theta_k}} \\ a\sim\pi_\theta}} \left[A^{\pi_{\theta_k}}(s,a)\right] \quad \text{s.t.} \quad \bar{D}_{\mathrm{KL}}(\pi_\theta \parallel \pi_{\theta_k}) \leq \delta \tag{15}$$

where $\bar{D}_{\mathrm{KL}}(\pi_\theta \parallel \pi_{\theta_k}) := \mathbb{E}_{s\sim d_{\pi_{\theta_k}}} \left[D_{\mathrm{KL}}\left(\pi_\theta \| \pi_{\theta_k}\right)[s]\right]$. The constraint set $\{\pi_\theta \in \Pi_\Theta : \bar{D}_{\mathrm{KL}}(\pi_\theta \parallel \pi_{\theta_k}) \leq \delta\}$ is called the *trust region set*. This problem can be regarded as an average reward variant of TRPO from Schulman et al. (2015). Note that the advantage function in (15) is the *average reward advantage function* introduced in Section 2. When we set $\pi_{\theta_{k+1}}$ to be the optimal solution to (15), $\pi_{\theta_{k+1}}$ can be shown to have the following performance guarantee:

Proposition 3. *Let $\pi_{\theta_{k+1}}$ be the optimal solution to* (15) *for some $\pi_{\theta_k} \in \Pi_\Theta$. The policy performance difference between $\pi_{\theta_{k+1}}$ and π_{θ_k} can be lower bounded by*

$$\rho(\pi_{\theta_{k+1}}) - \rho(\pi_{\theta_k}) \geq -\xi^{\pi_{\theta_{k+1}}} \sqrt{2\delta} \tag{16}$$

where $\xi^{\pi_{\theta_{k+1}}} = (\kappa^{\pi_{\theta_{k+1}}} - 1) \max_s \mathbb{E}_{a\sim\pi_{\theta_{k+1}}} \left|A^{\pi_{\theta_k}}(s,a)\right|$.

Proof. Since $\bar{D}_{\mathrm{KL}}(\pi_{\theta_k} \parallel \pi_{\theta_k}) = 0$, π_{θ_k} is a feasible solution. The objective value is 0 for $\pi_\theta = \pi_{\theta_k}$. The bound follows from (12) and (13) where the average KL is bounded by δ. □

Several algorithms have been proposed for efficiently solving the discounted version of (15): Schulman et al. (2015) and Wu et al. (2017) converts (15) into a convex problem via Taylor approximations; another approach is to first solve (15) in the nonparametric policy space and then project the result back into the parameter space (Abdolmaleki et al., 2018; Vuong et al., 2019). These algorithms can be adapted for the average reward case and are theoretically justified via Theorem 1 and Proposition 3. One notable difference compared to the discounted case is the estimation of the critic, as discussed in the next section and in the Appendix D.

5.2 IMPLEMENTATION

In this section, we discuss how the average reward version of the TRPO algorithm (Schulman et al., 2015) — which we will refer to as ATRPO — can be implemented in practice. Algorithm 2 provides a basic outline of the ATRPO algorithm.

Algorithm 2 Average Reward TRPO (ATRPO)

Initialize: Policy parameters θ_0, value net parameters ϕ_0, learning rate α.

1: **for** $k = 0, 1, 2, \cdots$ **do**
2: Collect a sample trajectory $\{s_t, a_t, s_{t+1}, r_t\},\ t = 1, \ldots, N$ from the environment using π_{θ_k}.
3: Calculate sample average reward of π_{θ_k} via $\rho = \frac{1}{N}\sum_{t=1}^{N} r_t$.
4: **for** $t = 1, 2, \ldots, N$ **do**
5: Get target $V_t^{\text{target}} = r_t - \rho + V_{\phi_k}(s_{t+1})$
6: Get advantage estimate $\hat{A}(s_t, a_t) = r_t - \rho + V_{\phi_k}(s_{t+1}) - V_{\phi_k}(s_t)$
7: Update critic by

$$\phi_{k+1} \leftarrow \phi_k - \alpha \nabla_\phi \mathcal{L}(\phi_k)$$

where

$$\mathcal{L}(\phi_k) = \frac{1}{2}\sum_{t=1}^{N} \left\| V_{\phi_k}(s_t) - V_t^{\text{target}} \right\|^2$$

8: Use $\hat{A}(s_t, a_t)$ to update θ_k using TRPO policy update (Schulman et al., 2015).

The major difference between the TRPO algorithm and the ATRPO algorithm is how the target for the critic and the advantage function are calculated. Importantly, simply letting $\gamma \to 1$ in TRPO does not lead to Algorithm 2. This subtle but important difference leads to a significant improvement in sample efficiency, as shown in the section on experimental results.

In Algorithm 2, for illustrative purposes, we use the average reward one-step bootstrapped estimate for the target of the critic and the advantage function. In practice, we instead use an average reward version of the Generalized Advantage Estimator (GAE) from Schulman et al. (2016). In short, GAE uses a tunable eligibility trace parameter λ to act as a trade-off between the Monte Carlo estimate and the bootstrapped estimate. In the Appendix D we provide more detail on how GAE can be generalized to the average reward case.

6 RELATED WORK

Dynamic programming algorithms for finding the optimal average reward policies have been well-studied (Howard, 1960; Blackwell, 1962; Veinott, 1966). In contrary to our method which is based on the policy gradient approach, several Q-learning-like algorithms for problems with unknown dynamics have been proposed, such as R-Learning (Schwartz, 1993), RVI Q-Learning (Abounadi et al., 2001), and CSV-Learning (Yang et al., 2016). Mahadevan (1996) conducted a thorough empirical analysis of the R-Learning algorithm. We note that much of the previous work on average reward RL focuses on the tabular setting without function approximations, and the theoretical properties of many of these Q-learning-based algorithm are not well understood (in particular R-learning). More recently, POLITEX updates policies using a Boltzmann distribution over the sum of action-value function estimates of the previous policies (Abbasi-Yadkori et al., 2019) and Wei et al. (2020) introduced a model-free algorithm for optimizing the average reward of weakly-communicating MDPs. Both methods are shown to have theoretical guarantees under the tabular setting.

For policy gradient methods, Baxter & Bartlett (2001) showed that if $1/(1-\gamma)$ is large compared to the mixing time of the Markov Chain induced by the MDP, then the gradient of $\rho_\gamma(\pi)$ can accurately approximate the gradient of $\rho(\pi)$. Kakade (2001) extended upon this result and provided an error bound on using an optimal discounted policy to maximize the average reward. In contrast, our work directly deals with using policy gradient methods for the average reward objective and provides theoretical guidance on the optimal step size for each policy update.

Policy improvement bounds have been extensively explored in the discounted case. The results from Schulman et al. (2015) is an extension of Kakade & Langford (2002) which restricted the policy class

to a mixture of policies. Pirotta et al. (2013) also proposed an alternative generalization to Kakade & Langford (2002). Achiam et al. (2017) improved upon Schulman et al. (2015) by replacing the maximum divergence with the average divergence.

7 EXPERIMENTS

Recently, DRL algorithms such as TRPO have proven to be successful for episodic high-dimensional tasks. In our experiments, we wish to study whether for continuing-control tasks, the policy trained with ATRPO can out-perform the policies trained with TRPO with different discount factors.

Our design goal for the experiments is to simulate continuing-control tasks where the agent can interact with the environment indefinitely. We consider three tasks (Ant, HalfCheetah, and Humanoid) from the MuJoCo physical simulator (Todorov et al., 2012) implemented in the OpenAI gym (Brockman et al., 2016). The natural goal is to train the agents to run as fast as possible without falling. However the standard MuJoCo tasks are episodic tasks which terminate when the agent falls. We convert these tasks into continuing control tasks via the following: when the agent falls, the agent incurs a large cost for falling, but then continues the trajectory from a random start state. We use these continuing-control tasks for both training and evaluation for both ATRPO and TRPO. More details on the environment can be found in Appendix F.

One point we wish to emphasize regarding the experiments is that even though the MuJoCo benchmark is commonly trained using the discounted objective (see e.g. Schulman et al. (2015), Wu et al. (2017), Schulman et al. (2017), Abdolmaleki et al. (2018), Vuong et al. (2019)), it is *always* evaluated using the undiscounted objective. This is because the undiscounted objective more naturally describes the goals of the MuJoCo agents (e.g., an agent's performance w.r.t. the reward signal should be equally important at time step 1000 as it is at time step 1). In the case of TRPO (and similarly many other DRL algorithms), discounting is used during training often for mathematical and computational convenience. Prior to our work, there has been no theoretical or empirical evidence to support applying trust region methods to the average reward. In this section, we demonstrate that when the actual objective we want to evaluate is undiscounted, discounting, as is commonly done, is unnecessary and may lead to suboptimal performance.

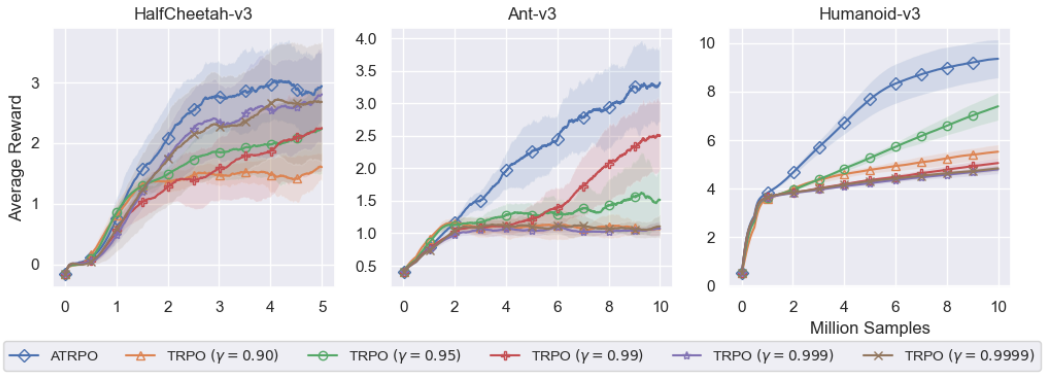

Figure 1: Learning curves comparing ATRPO and TRPO with different discount factors. The solid lines represent the average reward of trajectories of fixed length of 10,000 time steps averaged over the last 50 trajectories. The results are averaged over 10 random seeds and the shaded region represents one standard deviation.

During training, we collect one trajectory of a fixed length of 10,000 using the current policy.[1] We then use this data to update the critic and policy networks (see Algorithm 2). This gives us a new policy and critic which we then use to repeat the above process. In Figure 1, we plot the

[1]In the original OpenAI gym version of MuJoCo, episode lengths are capped at 1000 (see `https://github.com/openai/gym/blob/master/gym/envs/__init__.py`). We removed this cap to allow for arbitrarily long time horizons.

training curves of ATRPO and of TRPO for different discount factors. Detailed specifications and hyperparameter settings can be found in Appendix F.

Figure 1 shows that ATRPO improves performance by 5.0%, 32.8%, 26.7% on HalfCheetah, Ant and Humanoid respectively over TRPO with its best discount factors. One point worth noting is that increasing the discount factor does not necessarily lead to better performance of TRPO. A larger discount factor in principle enables the algorithm to seek a policy that performs well for the average-reward criterion. But, unfortunately, a larger discount factor can also increase the variance of the gradient estimator (Zhao et al., 2011; Schulman et al., 2016) and degrade generalization (Amit et al., 2020). Moreover, algorithms with discounting become unstable as $\gamma \to 1$ (Naik et al., 2019). The discount factor therefore serves as a hyperparameter which can be tuned to improve performance. This is supported by the observation that the optimal discount factor is different for each environment (0.999, 0.99, 0.95 for HalfCheetah, Ant, and Humanoid respectively), where choosing a suboptimal discount factor can have significant consequences. (For Ant and Humanoid, the optimal discount factor is 33.9% and 65.6% better than the second best discount factor.) We have shown here that using the average reward criterion not only delivers superior performance but also obviates the need to tune the discount factor.

To further support our conclusion, we will also compare ATRPO and TRPO using an alternative evaluation protocol. In this protocol, after every one million samples of training we run 10 separate evaluation trajectories of fixed length 10,000 time steps using the current policy with no exploration. The random seeds used for evaluation are different from those used in training. Figure 2 shows the average reward of these trajectories, Once again ATRPO provides superior performance.

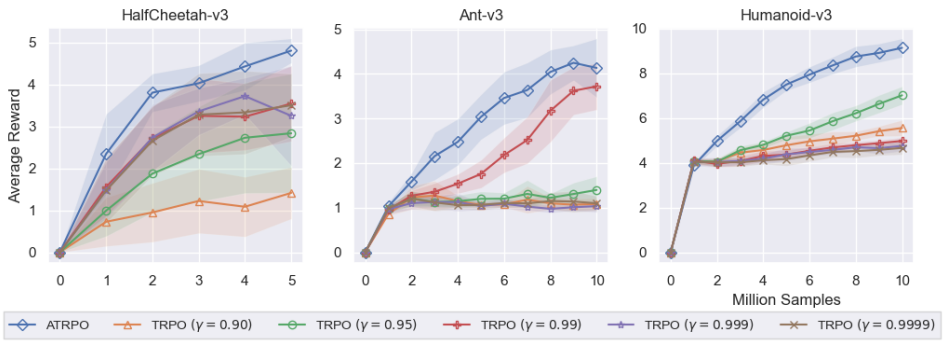

Figure 2: Comparing performance on evaluation trajectories of length 10,000. For each random seed used in training, we use a different unseen random seed to run 10 test trajectories after every 1 million samples of training. The solid line is averaged over these unseen random seeds. The shaded area is one standard deviation.

8 Conclusion

In this paper, we introduced a novel policy improvement bound for the average reward criterion. The bound is based on the average divergence between two policies and Kemeny's constant. We showed that previous existing policy improvement bounds for the discounted case results in a non-meaningful bound for the average reward objective. Our work provided the theoretical justification and the means to generalize the popular trust-region based algorithms to the average reward setting. We demonstrated through a series of experiments that our method is highly effective on high-dimensional continuing control tasks. In particular, we showed that when the natural objective of the task is undiscounted, discounting can lead to suboptimal behavior. To the best of our knowledge, we are one of the first to address how DRL methods can be used to learn undiscounted continuing control tasks with large state and action spaces.

REFERENCES

Yasin Abbasi-Yadkori, Peter Bartlett, Kush Bhatia, Nevena Lazic, Csaba Szepesvari, and Gellért Weisz. Politex: Regret bounds for policy iteration using expert prediction. In *International Conference on Machine Learning*, pp. 3692–3702, 2019.

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. *International Conference on Learning Representation (ICLR)*, 2018.

Jinane Abounadi, D Bertsekas, and Vivek S Borkar. Learning algorithms for markov decision processes with average cost. *SIAM Journal on Control and Optimization*, 40(3):681–698, 2001.

Joshua Achiam. UC Berkeley CS 285 (Fall 2017), Advanced Policy Gradients, 2017. URL: `http://rail.eecs.berkeley.edu/deeprlcourse-fa17/f17docs/lecture_13_advanced_pg.pdf`. Last visited on 2020/05/24.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 22–31. JMLR. org, 2017.

David Aldous and James Fill. Reversible markov chains and random walks on graphs, 1995.

Eitan Altman. *Constrained Markov decision processes*, volume 7. CRC Press, 1999.

Ron Amit, Ron Meir, and Kamil Ciosek. Discount factor as a regularizer in reinforcement learning. In *International conference on machine learning*, 2020.

Jonathan Baxter and Peter L Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.

Dimitri P Bertsekas, Dimitri P Bertsekas, Dimitri P Bertsekas, and Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1,2. Athena scientific Belmont, MA, 1995.

David Blackwell. Discrete dynamic programming. *The Annals of Mathematical Statistics*, pp. 719–726, 1962.

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

Grace E Cho and Carl D Meyer. Comparison of perturbation bounds for the stationary distribution of a markov chain. *Linear Algebra and its Applications*, 335(1-3):137–150, 2001.

Eyal Even-Dar, Sham M Kakade, and Yishay Mansour. Online markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009.

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. *International Conference on Machine Learning (ICML)*, 2018.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

Ronald A Howard. *Dynamic programming and markov processes.* John Wiley, 1960.

Jeffrey J Hunter. Stationary distributions and mean first passage times of perturbed markov chains. *Linear Algebra and its Applications*, 410:217–243, 2005.

Sham Kakade. Optimizing average reward using discounted rewards. In *International Conference on Computational Learning Theory*, pp. 605–615. Springer, 2001.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, volume 2, pp. 267–274, 2002.

L.C.M. Kallenberg. *Linear Programming and Finite Markovian Control Problems*. Centrum Voor Wiskunde en Informatica, 1983.

J.G. Kemeny and I.J. Snell. *Finite Markov Chains*. Van Nostrand, New Jersey, 1960.

Erich L Lehmann and George Casella. *Theory of point estimation*. Springer Science & Business Media, 2006.

Sridhar Mahadevan. Average reward reinforcement learning: Foundations, algorithms, and empirical results. *Machine learning*, 22(1-3):159–195, 1996.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *NIPS Deep Learning Workshop*, 2013.

Abhishek Naik, Roshan Shariff, Niko Yasui, and Richard S Sutton. Discounted reinforcement learning is not an optimization problem. *NeurIPS Optimization Foundations for Reinforcement Learning Workshop*, 2019.

Gergely Neu, Andras Antos, András György, and Csaba Szepesvári. Online markov decision processes under bandit feedback. In *Advances in Neural Information Processing Systems*, pp. 1804–1812, 2010.

Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural networks*, 21(4):682–697, 2008.

Matteo Pirotta, Marcello Restelli, Alessio Pecorino, and Daniele Calandriello. Safe policy iteration. In *International Conference on Machine Learning*, pp. 307–315, 2013.

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

Keith W Ross. Constrained markov decision processes with queueing applications. *Dissertation Abstracts International Part B: Science and Engineering[DISS. ABST. INT. PT. B- SCI. & ENG.],*, 46(4), 1985.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pp. 1889–1897, 2015.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *International Conference on Learning Representations (ICLR)*, 2016.

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

Anton Schwartz. A reinforcement learning method for maximizing undiscounted rewards. In *Proceedings of the tenth international conference on machine learning*, volume 298, pp. 298–305, 1993.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419): 1140–1144, 2018.

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*, pp. 1057–1063, 2000.

Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. *International Conference on Learning Representation (ICLR)*, 2019.

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

Alexandre B Tsybakov. *Introduction to nonparametric estimation*. Springer Science & Business Media, 2008.

Arthur F Veinott. On finding optimal policies in discrete dynamic programming with no discounting. *The Annals of Mathematical Statistics*, 37(5):1284–1294, 1966.

Quan Vuong, Yiming Zhang, and Keith W Ross. Supervised policy update for deep reinforcement learning. In *International Conference on Learning Representation (ICLR)*, 2019.

Chen-Yu Wei, Mehdi Jafarnia-Jahromi, Haipeng Luo, Hiteshi Sharma, and Rahul Jain. Model-free reinforcement learning in infinite-horizon average-reward markov decision processes. In *International conference on machine learning*, 2020.

Yuhuai Wu, Elman Mansimov, Roger B Grosse, Shun Liao, and Jimmy Ba. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. In *Advances in neural information processing systems (NIPS)*, pp. 5285–5294, 2017.

Shangdong Yang, Yang Gao, Bo An, Hao Wang, and Xingguo Chen. Efficient average reward reinforcement learning using constant shifting values. In *AAAI*, pp. 2258–2264, 2016.

Yiming Zhang, Quan Vuong, and Keith Ross. First order constrained optimization in policy space. *Advances in Neural Information Processing Systems*, 33, 2020.

Tingting Zhao, Hirotaka Hachiya, Gang Niu, and Masashi Sugiyama. Analysis and improvement of policy gradient estimation. In *Advances in Neural Information Processing Systems*, pp. 262–270, 2011.

A DISCOUNTED AND AVERAGE REWARD VALUE FUNCTIONS

The following result relates the discounted and average reward value functions.

Proposition A.1 (Blackwell (1962))**.** *For a given stationary policy π and discount factor $\gamma \in (0, 1)$,*

$$\lim_{\gamma \to 1} \left(V_\gamma^\pi(s) - \frac{\rho(\pi)}{1-\gamma} \right) = V^\pi(s) \tag{17}$$

for all $s \in \mathcal{S}$.

From Proposition A.1, it is clear that $\lim_{\gamma \to 1}(1-\gamma)\rho_\gamma(\pi) = \rho(\pi)$, i.e. the discounted and average reward objective are equivalent in the limit as γ approaches 1 . We can derive similar relations for the action-value function and advantage function.

Corollary A.1. *For a given stationary policy π and discount factor $\gamma \in (0, 1)$,*

$$\lim_{\gamma \to 1} \left(Q_\gamma^\pi(s,a) - \frac{\rho(\pi)}{1-\gamma} \right) = Q^\pi(s,a) \tag{18}$$

$$\lim_{\gamma \to 1} A_\gamma^\pi(s,a) = A^\pi(s,a) \tag{19}$$

for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$.

Proof. From Proposition A.1, we can rewrite (17) as

$$V_\gamma^\pi(s) = \frac{\rho(\pi)}{1-\gamma} + V^\pi(s) + g(\gamma, s) \tag{20}$$

where $\lim_{\gamma \to 1} g(\gamma, s) = 0$. We then expand $Q_\gamma^\pi(s,a)$ using the Bellman equation

$$\begin{aligned}
Q_\gamma^\pi(s,a) &= r(s,a) + \gamma \sum_{s'} P(s'|s,a) V_\gamma^\pi(s') \\
&= r(s,a) + \gamma \sum_{s'} P(s'|s,a) \left(\frac{\rho(\pi)}{1-\gamma} + V^\pi(s') + g^\pi(\gamma, s') \right) \\
&= r(s,a) + \frac{\gamma \rho(\pi)}{1-\gamma} + \gamma \sum_{s'} P(s'|s,a) \left(V^\pi(s') + g^\pi(\gamma, s') \right) \\
&= r(s,a) - \rho(\pi) + \frac{\rho(\pi)}{1-\gamma} + \sum_{s'} P(s'|s,a) V^\pi(s') \\
&- (1-\gamma) \sum_{s'} P(s'|s,a) V^\pi(s') + \gamma \sum_{s'} P(s'|s,a) g^\pi(\gamma, s') \\
&= Q^\pi(s,a) + \frac{\rho(\pi)}{1-\gamma} - (1-\gamma) \sum_{s'} P(s'|s,a) V^\pi(s') + \gamma \sum_{s'} P(s'|s,a) g^\pi(\gamma, s')
\end{aligned}$$

where we used Proposition A.1 for the second equality. Note that the last two terms in the last equality approach 0 as $\gamma \to 1$, rearranging the terms and taking the limit for $\gamma \to 1$ gives us Equation (18).

We can then similarly rewrite (18) as

$$Q_\gamma^\pi(s,a) = \frac{\rho(\pi)}{1-\gamma} + Q^\pi(s,a) + h(\gamma, s, a) \tag{21}$$

with $\lim_{\gamma \to 1} h(\gamma, s, a) = 0$. This allows us to rewrite the discounted advantage function as

$$\begin{aligned}
A_\gamma^\pi(s,a) &= Q_\gamma^\pi(s,a) - V_\gamma^\pi(s) \\
&= Q^\pi(s,a) + \frac{\rho(\pi)}{1-\gamma} + h^\pi(s,a,\gamma) - V^\pi(s) - \frac{\rho(\pi)}{1-\gamma} - g^\pi(s,\gamma) \\
&= A^\pi(s,a) + h^\pi(s,a,\gamma) - g^\pi(s,\gamma)
\end{aligned}$$

Since $h^\pi(s,a,\gamma)$ and $g^\pi(s,\gamma)$ both approach 0 as γ approaches 1, taking the limit for $\gamma \to 1$ gives us Equation (19). □

B PROOFS

B.1 PROOF OF PROPOSITION 1

Proposition 1. *Consider the following bound from Achiam et al. (2017)*

$$D^{-}_{\pi,\gamma}(\pi') \leq \rho_\gamma(\pi') - \rho_\gamma(\pi) \leq D^{+}_{\pi,\gamma}(\pi') \tag{4}$$

where

$$D^{\pm}_{\pi,\gamma}(\pi') = \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s\sim d_\pi \\ a\sim\pi}} \left[\frac{\pi'(a|s)}{\pi(a|s)} A^\pi_\gamma(s,a)\right] \pm \frac{2\gamma\epsilon_\gamma}{(1-\gamma)^2} \mathop{\mathbb{E}}_{s\sim d_\pi} [D_{TV}(\pi' \parallel \pi)[s]]$$

and $\epsilon_\gamma = \max_s \left|\mathbb{E}_{a\sim\pi'}[A^\pi_\gamma(s,a)]\right|$. *We have:*

$$\lim_{\gamma\to 1} (1-\gamma) D^{\pm}_{\pi,\gamma}(\pi') = \pm\infty \tag{5}$$

Proof. Since $d_{\pi,\gamma}$ approaches the stationary distribution d_π as $\gamma \to 1$, we can write the limit in (5) as

$$\begin{aligned}
&\lim_{\gamma\to 1} \left(\mathop{\mathbb{E}}_{\substack{s\sim d_{\pi,\gamma} \\ a\sim\pi'}} [A^\pi_\gamma(s,a)] \pm \frac{2\gamma\epsilon^\gamma}{1-\gamma} \mathop{\mathbb{E}}_{s\sim d_{\pi,\gamma}} D_{\text{TV}}(\pi' \parallel \pi) \right) \\
&= \mathop{\mathbb{E}}_{\substack{s\sim d_\pi \\ a\sim\pi'}} [A^\pi(s,a)] \pm 2\epsilon \mathop{\mathbb{E}}_{s\sim d_\pi} [D_{\text{TV}}(\pi' \parallel \pi)] \lim_{\gamma\to 1} \frac{\gamma}{1-\gamma} \\
&= \pm\infty
\end{aligned}$$

Here $\epsilon^\gamma = \max_s \left|\mathbb{E}_{a\sim\pi'}[A^\pi_\gamma(s,a)]\right|$ and $\epsilon = \max_s |\mathbb{E}_{a\sim\pi'}[A^\pi(s,a)]|$. The first equality is a direct result of Corollary A.1. By a similar argument, we can also show that the right-hand-side for Theorem 1 in (Schulman et al., 2015) also approaches infinity as γ approaches 1. □

B.2 PROOF OF LEMMA 1

Lemma 1. *For any two stochastic policies* π *and* π'*:*

$$\rho(\pi') - \rho(\pi) = \lim_{N\to\infty} \frac{1}{N} \mathop{\mathbb{E}}_{\tau\sim\pi'} \left[\sum_{t=0}^{N-1} A^\pi(s_t, a_t)\right] = \mathop{\mathbb{E}}_{\substack{s\sim d_{\pi'} \\ a\sim\pi'}} [A^\pi(s,a)] \tag{6}$$

Proof. We offer two approaches for this proof here. In the first approach, we directly expand the right-hand side using the definition of the advantage function and Bellman equation, which gives us:

$$\begin{aligned}
&\lim_{N\to\infty} \frac{1}{N} \mathop{\mathbb{E}}_{\tau\sim\pi'} \left[\sum_{t=0}^{N-1} A^\pi(s_t,a_t)\right] \\
=&\lim_{N\to\infty} \frac{1}{N} \mathop{\mathbb{E}}_{\tau\sim\pi'} \left[\sum_{t=0}^{N-1} (Q^\pi(s_t,a_t) - V^\pi(s_t))\right] \\
=&\lim_{N\to\infty} \frac{1}{N} \mathop{\mathbb{E}}_{\tau\sim\pi'} \left[\sum_{t=0}^{N-1} \left(r(s_t,a_t) - \rho(\pi) + \mathop{\mathbb{E}}_{s_{t+1}\sim P(\cdot|s_t,a_t)} [V^\pi(s_{t+1})] - V^\pi(s_t)\right)\right] \\
=&\lim_{N\to\infty} \frac{1}{N} \mathop{\mathbb{E}}_{\tau\sim\pi'} \left[\sum_{t=0}^{N-1} (r(s_t,a_t) - \rho(\pi) + V^\pi(s_{t+1}) - V^\pi(s_t))\right] \\
=&\rho(\pi') - \rho(\pi) + \lim_{N\to\infty} \frac{1}{N} \mathop{\mathbb{E}}_{\tau\sim\pi'} \left[\sum_{t=0}^{N-1} (V^\pi(s_{t+1}) - V^\pi(s_t))\right] \\
=&\rho(\pi') - \rho(\pi) + \lim_{N\to\infty} \frac{1}{N} \mathop{\mathbb{E}}_{\tau\sim\pi'} [V^\pi(s_N) - V^\pi(s_0)] \\
=&\rho(\pi') - \rho(\pi).
\end{aligned}$$

The last equality can be obtained by rewriting the expectation

$$\begin{aligned}\lim_{N\to\infty}\frac{1}{N}\mathop{\mathbb{E}}_{\tau\sim\pi'}\left[\sum_{t=0}^{N-1}A^{\pi}(s_t,a_t)\right] &= \lim_{N\to\infty}\frac{1}{N}\left[\sum_{t=0}^{N-1}\sum_{s,a}P(s_t=s|\pi')\pi'(a|s)A^{\pi}(s,a)\right]\\ &= \sum_{s,a}\pi'(a|s)A^{\pi}(s,a)\lim_{N\to\infty}\frac{1}{T}\sum_{t=0}^{N-1}P(s_t=s|\pi')\\ &= \sum_{s,a}d^{\pi'}(s)\pi'(a|s)A^{\pi}(s,a) = \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi'}\\ a\sim\pi'}}[A^{\pi}(s,a)]\end{aligned}$$

Alternatively, we can directly apply Proposition A.1 and Corollary A.1 to Lemma 6.1 of (Kakade & Langford, 2002) and take the limit as $\gamma \to 1$. □

B.3 Proof of Lemma 2

Lemma 2. *For any two stochastic policies π and π', the following bound holds:*

$$\mathop{\mathbb{E}}_{\substack{s\sim d_{\pi}\\ a\sim\pi'}}[A^{\pi}(s,a)] - 2\epsilon D_{TV}(d_{\pi'}\parallel d_{\pi}) \leq \rho(\pi') - \rho(\pi) \leq \mathop{\mathbb{E}}_{\substack{s\sim d_{\pi}\\ a\sim\pi'}}[A^{\pi}(s,a)] + 2\epsilon D_{TV}(d_{\pi'}\parallel d_{\pi}) \quad (7)$$

where $\epsilon = \max_s \left|\mathbb{E}_{a\sim\pi'(a|s)}[A^{\pi}(s,a)]\right|$.

Proof.

$$\begin{aligned}\left|\rho(\pi') - \rho(\pi) - \mathop{\mathbb{E}}_{\substack{s\sim d_{\pi}\\ a\sim\pi'}}[A^{\pi}(s,a)]\right| &= \left|\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi'}\\ a\sim\pi'}}[A^{\pi}(s,a)] - \mathop{\mathbb{E}}_{\substack{s\sim d_{\pi}\\ a\sim\pi'}}[A^{\pi}(s,a)]\right|\\ &= \left|\sum_s \mathop{\mathbb{E}}_{a\sim\pi'}[A^{\pi}(s,a)]\left(d_{\pi'}(s) - d_{\pi}(s)\right)\right|\\ &\leq \sum_s \left|\mathop{\mathbb{E}}_{a\sim\pi'}[A^{\pi}(s,a)]\left(d_{\pi'}(s) - d_{\pi}(s)\right)\right|\\ &\leq \max_s \left|\mathop{\mathbb{E}}_{a\sim\pi'}[A^{\pi}(s,a)]\right| \|d_{\pi'} - d_{\pi}\|_1\\ &= 2\epsilon D_{\text{TV}}(d^{\pi'}\parallel d^{\pi})\end{aligned}$$

where the last inequality follows from Hölder's inequality. □

B.4 Proof of Lemma 3

Lemma 3. *The divergence between the stationary distributions d_{π} and $d_{\pi'}$ can be upper bounded by the average divergence between policies π and π' as follows:*

$$D_{TV}(d_{\pi'}\parallel d_{\pi}) \leq (\kappa^{\pi'} - 1)\mathop{\mathbb{E}}_{s\sim d_{\pi}}[D_{TV}(\pi'\parallel\pi)[s]] \quad (10)$$

Proof. Our proof is based on Markov chain perturbation theory (Cho & Meyer, 2001; Hunter, 2005). Note first that

$$\begin{aligned}(d_{\pi'}^T - d_{\pi}^T)(I - P_{\pi'} + P_{\pi'}^*) &= d_{\pi'}^T - d_{\pi}^T - d_{\pi'}^T + d_{\pi}^T P_{\pi'}\\ &= d_{\pi}^T P_{\pi'} - d_{\pi}^T \quad (22)\\ &= d_{\pi}^T(P_{\pi'} - P_{\pi})\end{aligned}$$

Right multiplying (22) by $(I - P_{\pi'} + P_{\pi'}^*)^{-1}$ gives us:

$$d_{\pi'}^T - d_{\pi}^T = d_{\pi}^T(P_{\pi'} - P_{\pi})(I - P_{\pi'} + P_{\pi'}^*)^{-1} \quad (23)$$

Recall that $Z^{\pi'} = (I - P_{\pi'} + P_{\pi'}^*)^{-1}$ and $M^{\pi'} = (I - Z^{\pi'} + EZ_{\text{dg}}^{\pi'})D^{\pi'}$. Rearranging the terms we find that

$$Z^{\pi'} = I + EZ_{\text{dg}}^{\pi'} - M^{\pi'}(D^{\pi'})^{-1} \quad (24)$$

Plugging (24) into (23) gives us

$$\begin{aligned} d_{\pi'}^T - d_\pi^T &= d_\pi^T (P_{\pi'} - P_\pi)(I + EZ_{\text{dg}}^{\pi'} - M^{\pi'}(D^{\pi'})^{-1}) \\ &= d_\pi^T (P_{\pi'} - P_\pi)(I - M^{\pi'}(D^{\pi'})^{-1}) \end{aligned} \tag{25}$$

where the last equality is due to $(P_{\pi'} - P_\pi)E = 0$.

By the submultiplicative property of operator norms (Horn & Johnson, 2012), we have:

$$\begin{aligned} \|d_{\pi'} - d_\pi\|_1 &= \left\| (I - M^{\pi'}(D^{\pi'})^{-1})^T (P_{\pi'}^T - P_\pi^T) d_\pi \right\|_1 \\ &\leq \left\| (I - M^{\pi'}(D^{\pi'})^{-1})^T \right\|_1 \left\| (P_{\pi'}^T - P_\pi^T) d_\pi \right\|_1 \\ &= \left\| (I - M^{\pi'}(D^{\pi'})^{-1}) \right\|_\infty \left\| (P_{\pi'}^T - P_\pi^T) d_\pi \right\|_1 \end{aligned} \tag{26}$$

We can rewrite $\left\| I - M^{\pi'}(D^{\pi'})^{-1} \right\|_\infty$ as

$$\begin{aligned} \left\| I - M^{\pi'}(D^{\pi'})^{-1} \right\|_\infty &= \max_s \left(\sum_{s'} M^{\pi'}(s, s') d_{\pi'}(s') - 1 \right) \\ &= \kappa^{\pi'} - 1 \end{aligned} \tag{27}$$

Finally we bound $\left\| (P_{\pi'}^T - P_\pi^T) d_\pi \right\|_1$ by

$$\begin{aligned} \left\| (P_{\pi'}^T - P_\pi^T) d_\pi \right\|_1 &= \sum_{s'} \left| \sum_s \left(\sum_a P(s'|s,a)\pi'(a|s) - P(s'|s,a)\pi(a|s) \right) d_\pi(s) \right| \\ &\leq \sum_{s',s} \left| \sum_a P(s'|s,a)(\pi'(a|s) - \pi(a|s)) \right| d_\pi(s) \\ &\leq \sum_{s,s',a} P(s'|s,a) \left| \pi'(a|s) - \pi(a|s) \right| d_\pi(s) \\ &\leq \sum_{s,a} \left| \pi'(a|s) - \pi(a|s) \right| d_\pi(s) \\ &= 2 \mathop{\mathbb{E}}_{s \sim d^\pi} [D_{\text{TV}}(\pi' \parallel \pi)] \end{aligned} \tag{28}$$

Plugging back into (26) gives the desired result. □

C Reinforcement Learning with Average Cost Constraints

In addition to learning to improve its long-term performance, many real-life applications of RL also require the agent to satisfy certain safety constraints. A mathematically principled framework for incorporating safety constraints into RL is using Constraint Markov Decision Processes (CMDP). A CMDP (Kallenberg, 1983; Ross, 1985; Altman, 1999) is an MDP equipped with a constraint set Π_c, a CMDP problems finds a policy π that maximizes an agent's long-run reward given that $\pi \in \Pi_c$. We consider two forms of constraint sets: the average cost constraint set $\{\pi \in \Pi : \rho_c(\pi) \leq b\}$ and the discounted cost constraint set $\{\pi \in \Pi : \rho_{c,\gamma}(\pi) \leq b\}$. Here b is some given constraint bound, $\rho_c(\pi) = \lim_{N\to\infty} \frac{1}{N} \mathbb{E}_{\tau\sim\pi} \left[\sum_{t=0}^{N-1} c(s_t, a_t) \right]$, and $\rho_{c,\gamma}(\pi) := \mathbb{E}_{\tau\sim\pi} \left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \right]$ for some bounded cost function $c : \mathcal{S} \times \mathcal{A} \to [c_{\min}, c_{\max}]$. However, directly adding cost constraints to any iterative policy improvement algorithms can be sample inefficient since the cost constraint needs to be evaluated using samples from the new policy after every policy update. Instead, Achiam et al. (2017) proposed updating π_{θ_k} via the following optimization problem:

$$\underset{\pi_\theta \in \Pi_\theta}{\text{maximize}} \quad \mathop{\mathbb{E}}_{\substack{s \sim d_{\pi_{\theta_k}, \gamma} \\ a \sim \pi_\theta}} [A_\gamma^{\pi_{\theta_k}}(s, a)] \quad \text{s.t.} \quad \tilde{\rho}_{c,\gamma}(\pi_\theta) \leq b, \; \bar{D}_{\text{KL}}(\pi_\theta \parallel \pi_{\theta_k}) \leq \delta. \tag{29}$$

Here, $\tilde{\rho}_{c,\gamma}(\pi_\theta) := \rho_{c,\gamma}(\pi_{\theta_k}) + \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{\pi_{\theta_k},\gamma},a\sim\pi_\theta}\left[A^{\pi_{\theta_k}}_{c,\gamma}(s,a)\right]$ and $A^{\pi_{\theta_k}}_{c,\gamma}(s,a)$ is the discounted cost advantage function where we replace the reward with the cost. Note that in (29) the original cost constraint was replaced by a discounted surrogate cost constraint $\tilde{\rho}_{c,\gamma}(\pi_\theta)$ which can be evaluated using samples from π_{θ_k}. The bound (4) still hold when the reward function is replaced with the cost (see also Corollary 2 from Achiam et al. (2017)). Therefore by (4) and (13):

$$|\rho_{c,\gamma}(\pi_\theta) - \tilde{\rho}_{c,\gamma}(\pi_\theta)| \leq \frac{\gamma\epsilon_{c,\gamma}}{(1-\gamma)^2}\sqrt{2\bar{D}_{\text{KL}}(\pi_\theta \parallel \pi_{\theta_k})} \tag{30}$$

where $\epsilon_{c,\gamma} = \max_s \left|\mathbb{E}_{a\sim\pi'}[A^{\pi}_{c,\gamma}(s,a)]\right|$. This shows that the surrogate cost is a good approximation to $\rho_{c,\gamma}(\pi_\theta)$ when π_θ and π_{θ_k} are close. Using (30) and the trust region constraint, Achiam et al. (2017) upper bounded the worst-case constraint violation for when $\pi_{\theta_{k+1}}$ is the solution to (29).

The framework is problematic when the cost constraint is undiscounted. Define the average surrogate cost as $\tilde{\rho}_c(\pi_\theta) := \rho_c(\pi_{\theta_k}) + \mathbb{E}_{s\sim d_{\pi_{\theta_k}},a\sim\pi_{\theta_k}}[A^{\pi_\theta}_c(s,a)]$. We can easily show that

$$\lim_{\gamma\to 1}(1-\gamma)(\rho_{c,\gamma}(\pi_\theta) - \tilde{\rho}_{c,\gamma}(\pi_\theta)) = \rho_c(\pi_\theta) - \tilde{\rho}_c(\pi_\theta) \quad \text{and} \quad \lim_{\gamma\to 1}\frac{\gamma\epsilon_{c,\gamma}}{1-\gamma}\sqrt{2\bar{D}_{\text{KL}}(\pi_\theta \parallel \pi_{\theta_k})} = \infty$$

However, by Theorem 1 and (13):

$$|\rho_c(\pi_\theta) - \tilde{\rho}_c(\pi_\theta)| \leq \xi^{\pi_\theta}_c\sqrt{2\bar{D}_{\text{KL}}(\pi_\theta \parallel \pi_{\theta_k})} \tag{31}$$

where $\xi^{\pi_\theta}_c = \max_s \mathbb{E}_{a\sim\pi_\theta}|A^{\pi_{\theta_k}}_c(s,a)|\left\|(I - P_{\pi_\theta} + P^*_{\pi_\theta})^{-1}\right\|_\infty$. We then have the following result:

Proposition C.1. *Suppose π_θ and π_{θ_k} satisfy the constraints $\tilde{\rho}_c(\pi_\theta) < b$ and $\bar{D}_{KL}(\pi_\theta \parallel \pi_{\theta_k}) \leq \delta$, then*

$$\rho_C(\pi_\theta) \leq b + \xi^{\pi_\theta}_c\sqrt{2\delta} \tag{32}$$

The upper-bound in Proposition C.1 provides a worst-case constraint violation guarantee when π_θ is the solution to the average-cost variant of (29). It is an undiscounted parallel to Proposition 2 in Achiam et al. (2017) which provides a similar guarantee for the discounted case. It shows that contrary to what was previously believed (Tessler et al., 2019), (29) can easily be modified to accommodate for average cost constraints and still satisfy an upper bound for worst-case constraint violation. Scalable algorithms have been proposed for approximately solving (29) (Achiam et al., 2017; Zhang et al., 2020). Proposition C.1 shows that these algorithms can be generalized to average cost constraints with only minor modifications. In Appendix E.2, we will show how the CPO algorithm (Achiam et al., 2017) can be modified for average cost constraints.

D Critic Estimation for the Average Reward Setting

Suppose the agent collects a batch of data consisting of a trajectories each of length N $\{s_t, a_t, r_t, s_{t+1}\}$ $(t = 1, \ldots, N)$ using policy π. Similar to what is commonly done for critic estimation in on-policy methods, we fit some value function $\hat{V}^\pi_\phi$ parameterized by ϕ using data collected with the policy.

We will first review how this is done in the discounted case. Two of the most common ways of calculating the regression target for $\hat{V}^\pi_\phi$ are the *Monte Carlo* target denoted by

$$V^{\text{target}}_t = \sum_{t'=t}^{N}\gamma^{t'-t}r_t, \tag{33}$$

or the *bootstrapped* target

$$V^{\text{target}}_t = r_t + \gamma\hat{V}^\pi_\phi(s_{t+1}). \tag{34}$$

Using the dataset $\{s_t, y_t\}$, we can fit $\hat{V}^\pi_\phi$ with supervised regression by minimizing the MSE between $\hat{V}^\pi_\phi(s_t)$ and y_t. With the fitted value function, we can estimate the advantage function either with the Monte Carlo estimator

$$\hat{A}^\pi_{\text{MC}}(s_t, a_t) = \sum_{t'=t}^{N}\gamma^{t'-t}r_t - \hat{V}^\pi_\phi(s_t)$$

or the bootstrap estimator

$$\hat{A}^{\pi}_{\text{BS}}(s_t, a_t) = r_t + \gamma \hat{V}^{\pi}_{\phi}(s_{t+1}) - \hat{V}^{\pi}_{\phi}(s_t).$$

When the Monte Carlo advantage estimator is used to approximate the policy gradient, it does not introduce a bias but tends to have a high variance whereas the bootstrapped estimator introduces a bias but tends to have lower variance. These two estimators are seen as the two extreme ends of the bias-variance trade-off. In order to have better control over the bias and variance, Schulman et al. (2016) used the idea of eligibility traces (Sutton & Barto, 2018) and introduced the Generalized Advantage Estimator (GAE). The GAE takes the form

$$\hat{A}_{\text{GAE}}(s_t, a_t) = \sum_{t'=t}^{N} (\gamma\lambda)^{t'-t} \delta_{t'} \tag{35}$$

where

$$\delta_{t'} = r_{t'} + \gamma \hat{V}^{\pi}_{\phi}(s_{t'+1}) - \hat{V}^{\pi}_{\phi}(s_{t'}) \tag{36}$$

and $\lambda \in [0, 1]$ is the eligibility trace parameter. We can then use the parameter λ to tune the bias-variance trade-off. It is worth noting two special cases corresponding to the bootstrap and Monte Carlo estimator:

$$\lambda = 0: \quad \hat{A}_{\text{GAE}}(s_t, a_t) = r_t + \gamma \hat{V}^{\pi}_{\phi}(s_{t+1}) - \hat{V}^{\pi}_{\phi}(s_t)$$

$$\lambda = 1: \quad \hat{A}_{\text{GAE}}(s_t, a_t) = \sum_{t'=t}^{N} \gamma^{t'-t} r_{t'} - \hat{V}^{\pi}_{\phi}(s_t)$$

For infinite horizon tasks, the discount factor γ is used to reduce variance by downweighting rewards far into the future (Schulman et al., 2016). Also noted in (Schulman et al., 2016) is that for any $l \gg 1/(1-\gamma)$, γ^l decreases rapidly and any effects resulting from actions after $l \approx 1/(1-\gamma)$ are effectively "forgotten". This approach in essence converts a continuous control task into an episodic task where any rewards received after $l \approx 1/(1-\gamma)$ becomes negligible. This undermines the original continuing nature of the task and could prove to be especially problematic for problems where effects of actions are delayed far into the future. However, increasing γ would lead to an increase in variance. Thus in practice γ is often treated as a hyperparameter to balance the effective horizon of the task and the variance of the gradient estimator.

To mitigate this, we introduce how we can formulate critics for the average reward. A key difference between the two cases is that in the discounted case we use $\hat{V}^{\pi}_{\phi}$ to approximate the *discounted value function* whereas in the average reward case $\hat{V}^{\pi}_{\phi}$ is used to approximate the *average value function* (recall definition from Section 2).

Let

$$\hat{\rho}_{\pi} = \frac{1}{N} \sum_{t=1}^{N} r_t$$

denote the estimated average reward. The Monte Carlo target for the average reward value function is

$$V_t^{\text{target}} = \sum_{t'=t}^{N} (r_t - \hat{\rho}_{\pi}) \tag{37}$$

and the bootstrapped target is

$$V_t^{\text{target}} = r_t - \hat{\rho}_{\pi} + \hat{V}^{\pi}_{\phi}(s_{t+1}). \tag{38}$$

Note that our targets (37-38) are distinctly different from the traditional discounted targets (33-34).

The Monte Carlo and Bootstrap estimators for the average reward advantage function are:

$$\hat{A}^{\pi}_{\text{MC}}(s_t, a_t) = \sum_{t'=t}^{N} (r_t - \hat{\rho}_{\pi}) - \hat{V}^{\pi}_{\phi}(s_t)$$

$$\hat{A}^{\pi}_{\text{BS}}(s_t, a_t) = r_{i,t} - \hat{\rho}_{\pi} + \hat{V}^{\pi}_{\phi}(s_{t+1}) - \hat{V}^{\pi}_{\phi}(s_t)$$

We can similarly extend the GAE to the average reward setting:

$$\hat{A}_{\text{GAE}}(s_t, a_t) = \sum_{t'=t}^{N} \lambda^{t'-t} \delta_{t'} \tag{39}$$

where

$$\delta_{t'} = r_{t'} - \hat{\rho}_\pi + \hat{V}_\phi^\pi(s_{t'+1}) - \hat{V}_\phi^\pi(s_{t'}). \tag{40}$$

and set the target for the value function to

$$V_t^{\text{target}} = r_t - \hat{\rho}_\pi + \hat{V}_\phi^\pi(s_{t+1}) + \sum_{t'=t+1}^{N} \lambda^{t'-t} \delta_{t'} \tag{41}$$

The two special cases corresponding to $\lambda = 0$ and $\lambda = 1$ are

$$\begin{aligned} \lambda = 0: &\quad \hat{A}_{\text{GAE}}(s_t, a_t) = r_t - \hat{\rho}_\pi + \hat{V}_\phi^\pi(s_{t+1}) - \hat{V}_\phi^\pi(s_t) \\ \lambda = 1: &\quad \hat{A}_{\text{GAE}}(s_t, a_t) = \sum_{t'=t}^{N} (r_{t'} - \hat{\rho}_\pi) - \hat{V}_\phi^\pi(s_t) \end{aligned}$$

We note again that the average reward advantage estimator is distinct from the discounted case. To summarize, in the average reward setting:

- The parameterized value function is used to fit the *average reward* value function.
- The reward term r_t in the discounted formulation is replaced by $r_t - \hat{\rho}_\pi$.
- Without any discount factors, recent and future experiences are weighed equally thus respecting the continuing nature of the task.

E Algorithmic Details

E.1 ATRPO

The ATRPO algorithm is almost identical to its discounted counterpart (Schulman et al., 2015) with the exception of the critic estimation (see Appendix D). Here we give a brief summary of the algorithm, for more details see Schulman et al. (2015) or the lecture notes from Achiam (2017).

We approximate (15) using Taylor approximations

$$\begin{aligned} \underset{\theta}{\text{maximize}} \quad & g^T(\theta - \theta_k) \\ \text{subject to} \quad & \frac{1}{2}(\theta - \theta_k)^T H (\theta - \theta_k) \leq \delta \end{aligned} \tag{42}$$

where

$$g := \mathop{\mathbb{E}}_{\substack{s \sim d_{\pi_{\theta_k}} \\ a \sim \pi_{\theta_k}}} \left[\nabla_\theta \log \pi_\theta(a|s)|_{\theta=\theta_k} A^{\pi_{\theta_k}}(s, a) \right] \tag{43}$$

and

$$H := \mathop{\mathbb{E}}_{\substack{s \sim d_{\pi_{\theta_k}} \\ a \sim \pi_{\theta_k}}} \left[\nabla_\theta \log \pi_\theta(a|s)|_{\theta=\theta_k} \nabla_\theta \log \pi_\theta(a|s)|_{\theta=\theta_k}^T \right] \tag{44}$$

Note that g is identical to the average reward policy gradient (Sutton et al., 2000) and H is known as the Fisher Information Matrix (FIM) (Lehmann & Casella, 2006). The solution to (42) is

$$\theta = \theta_k + \sqrt{\frac{2\delta}{g^T H^{-1} g}} H^{-1} g \tag{45}$$

For policy classes with high-dimensional parameter spaces, estimating the inverse of H can be computationally challenging. Like in (Schulman et al., 2015), we use the conjugate gradient method to approximate H. Finally due to approximation errors, the update rule in (45) does not necessarily guarantee trust-region constraint satisfaction, therefore an exponential-decaying backtracking line search procedure is performed on the new parameter to ensure trust-region satisfaction.

E.2 ACPO

Like in the previous section, we will give a brief summary of the algorithm, more details can be found in (Achiam et al., 2017).

Using Taylor approximations, (29) can be written as

$$\begin{aligned} \underset{\theta}{\text{maximize}} \quad & g^T(\theta - \theta_k) \\ \text{subject to} \quad & \tilde{c} + \tilde{g}^T(\theta - \theta_k) \leq 0 \\ & \frac{1}{2}(\theta - \theta_k)^T H(\theta - \theta_k) \leq \delta \end{aligned} \tag{46}$$

where g, H were defined in the previous section, $\tilde{c} = \rho(\pi_{\theta_k}) - b$, and $\tilde{g}$ is the gradient of the constraint. This is a convex optimization problem where strong duality holds, hence it can be solved using a simple Lagrangian argument. The update rule takes the form

$$\theta = \theta_k + \frac{1}{\lambda} H^{-1}(g - \nu\tilde{g}) \tag{47}$$

where λ and ν are Lagrange multipliers satisfying

$$\max_{\lambda,\nu \geq 0} -\frac{1}{2\lambda}\left(g^T H^{-1} g + 2\nu g^T H^{-1}\tilde{g} + \nu^2 \tilde{g} H^{-1}\tilde{g}^T\right) + \nu\tilde{c} - \frac{1}{2}\lambda\delta \tag{48}$$

The dual problem (48) can be solved explicitly (Achiam et al., 2017). Similar to ATRPO, we use the conjugate gradient method to estimate H and perform a backtracking line search procedure to guarantee approximate constraint satisfaction.

F Experimental Details

All experiments were implemented in Pytorch 1.3.1 and Python 3.7.4 on Intel Xeon Gold 6230 processors. We based our TRPO implementation on `https://github.com/ikostrikov/pytorch-trpo` and `https://github.com/Khrylx/PyTorch-RL`. Our CPO implementation is our own Pytorch implementation based on `https://github.com/jachiam/cpo`. Our hyperparameter selections were also based on these implementations. Our choice of hyperparameters were based on the motivation that we wanted to put discounted TRPO in the best possible light and compare its performance with ATRPO. Our hyperparameter choices for ATRPO mirrored the discounted case since we wanted to understand how performance for the average reward case differs while controlling for all other variables.

We set the reset cost of 100 on all three environments. In the OpenAI Gym API, an agent at some current state receives an action from some policy, the API gives the next state of the agent, the reward, and a done signal which indicates whether the agent has reached the terminal state. When the agent falls (i.e. it receives a `done=True` signal), we subtract 100 from the reward received by the agent, and reset the next state using the `reset()` method from the API. For more information on the Gym API, see `https://gym.openai.com/`.

We used a two-layer feedforward neural network with a $\tanh$ activation for both our policy and value networks. The policy is Gaussian with a diagonal covariance matrix. The policy networks outputs a mean vector and a vector containing the state-independent log standard deviations. States are normalized by the running mean and the running standard deviation before being fed to any network. We used the GAE for advantage estimation (see Appendix D). The advantage values are normalized by the batch mean and batch standard deviation before being used for policy updates. Learning rates are linearly annealed to 0 over the course of training. Table 1 summarizes the hyperparameters used in our experiments.

Table 1: Hyperparameter Setup

Hyperparameter	ATRPO/TRPO
No. of hidden layers	2
No. of hidden nodes	64
Activation	$\tanh$
Initial log std	-1
Batch size	10,000
Training trajectory length	10,000
GAE parameter	0.95
Learning rate for policy	3×10^{-4}
Learning rate for value net	3×10^{-4}
$L2$-regularization coeff. for value net	3×10^{-3}
Damping coeff.	0.01
Backtracking coeff.	0.8
Max backtracking iterations	10
Max conjugate gradient iterations	10
Trust region bound δ	0.01