On-Policy Deep Reinforcement Learning for the Average-Reward Criterion

Yiming Zhang [1] **Keith W. Ross** [2 1]

Abstract

We develop theory and algorithms for average-reward on-policy Reinforcement Learning (RL). We first consider bounding the difference of the long-term average reward for two policies. We show that previous work based on the discounted return (Schulman et al., 2015; Achiam et al., 2017) results in a non-meaningful bound in the average-reward setting. By addressing the average-reward criterion directly, we then derive a novel bound which depends on the average divergence between the two policies and Kemeny's constant. Based on this bound, we develop an iterative procedure which produces a sequence of monotonically improved policies for the average reward criterion. This iterative procedure can then be combined with classic DRL (Deep Reinforcement Learning) methods, resulting in practical DRL algorithms that target the long-run average reward criterion. In particular, we demonstrate that Average-Reward TRPO (ATRPO), which adapts the on-policy TRPO algorithm to the average-reward criterion, significantly outperforms TRPO in the most challenging MuJuCo environments.

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), where the episode terminates when the game ends. In continuing tasks, such as robotic locomotion (Peters & Schaal, 2008; Schulman et al., 2015; Haarnoja et al., 2018) or in a queuing scenario (Tadepalli & Ok, 1994; Sutton & Barto, 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 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* so that the infinite-horizon return 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 is not appropriate when the goal is to optimize long-term behavior, i.e., when the natural objective underlying the task at hand is non-discounted. In particular, we note that for the vast majority of benchmarks for reinforcement learning such as Atari games (Mnih et al., 2013) and MuJoCo (Todorov et al., 2012), a non-discounted performance measure is used to evaluate the trained policies.

Although in many circumstances, non-discounted criteria are more natural, most of the successful DRL algorithms today have been designed to optimize a discounted criterion during training. One possible work-around for this mismatch is to simply train with a discount factor that is very close to one. Indeed, from the Blackwell optimality theory of MDPs (Blackwell, 1962), we know that if the discount factor is very close to one, then an optimal policy for the infinite-horizon discounted criterion is also optimal for the long-run average-reward criterion. However, although Blackwell's result suggests we can simply use a large discount factor to optimize non-discounted criteria, problems with large discount factors are in general more difficult to solve (Petrik & Scherrer, 2008; Jiang et al., 2015; 2016; Lehnert et al., 2018). Researchers have also observed that state-of-the-art DRL algorithms typically break down when the discount factor gets too close to one (Schulman et al., 2016; Andrychowicz et al., 2020).

In this paper we seek to develop algorithms for finding high-performing policies for average-reward DRL problems. Instead of trying to simply use standard discounted DRL algorithms with large discount factors, we instead attack the problem head-on, seeking to directly optimize the average-reward criterion. While the average reward setting has been

[1]New York University [2]New York University Shanghai. Correspondence to: Yiming Zhang <yiming.zhang@cs.nyu.edu>.

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

extensively studied in the classical Markov Decision Process literature (Howard, 1960; Blackwell, 1962; Veinott, 1966; Bertsekas et al., 1995), and has to some extent been studied for tabular RL (Schwartz, 1993; Mahadevan, 1996; Abounadi et al., 2001; Wan et al., 2020), it has received relatively little attention in the DRL community. In this paper, our focus is on developing average-reward on-policy DRL algorithms.

One major source of difficulty with modern on-policy 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' by building upon the work of Kakade & Langford (2002). 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 for 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 long-run rewards. The bound depends on the average divergence between the policies and on the so-called Kemeny constant, which measures to what degree the irreducible Markov chains associated with the policies are "well-mixed". 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. In particular, we describe in detail the Average Reward TRPO (ATRPO) algorithm, which is the average reward variant of the TRPO algorithm (Schulman et al., 2015). Using the MuJoCo simulated robotic benchmark, we carry out extensive experiments demonstrating the effectiveness of of ATRPO compared to its discounted counterpart, in particular on the most challenging MuJoCo tasks. Notably, we show that ATRPO can significantly out-perform TRPO on a set of high-dimensional continuing control tasks.

Our main contributions can be summarized as follows:

- We extend the policy improvement bound from Schulman et al. (2015) and Achiam et al. (2017) to the average reward setting. We demonstrate that our new bound depends on the average divergence between the two policies and on the mixing time of the underlying Markov chain.
- We use the aforementioned policy improvement bound to derive novel on-policy deep reinforcement learning algorithms for optimizing the average reward.
- Most modern DRL algorithms introduce a discount factor during training even when the natural objective of interest is undiscounted. This leads to a discrepancy between the evaluation and training objective. We demonstrate that optimizing the average reward directly can effectively address this mismatch and lead to much stronger performance.

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 $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ be a stationary policy where $\Delta(\mathcal{A})$ is the probabilty simplex over $\mathcal{A}$, and Π is the set of all stationary policies. We consider two classes of MDPs:

Assumption 1 (Ergodic)**.** *For every stationary policy, the induced Markov chain is irreducible and aperiodic.*

Assumption 2 (Aperiodic Unichain)**.** *For every stationary policy, the induced Markov chain contains a single aperiodic recurrent class and a finite but possibly empty set of transient states.*

By definition, any MDP which satisfies Assumption 1 is also unichain. We note that most MDPs of practical interest belong in these two classes. We will mostly focus on MDPs which satisfy Assumption 1 in the main text. In the supplementary material, we will address the aperiodic unichain case. Here we present the two objective formulations for continuing control tasks: the average reward approach and discounted reward criterion.

Average Reward Criterion

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 $d_\pi(s) := \lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-1} P_{\tau\sim\pi}(s_t = s)$ is the *stationary state distribution under policy* π, and $\tau = (s_0, a_0, \dots,)$ is a sample trajectory. The limits in $\rho(\pi)$ and $d_\pi(s)$ are guaranteed to exist under our assumptions. Since the MDP is aperiodic, it can also be shown that $d_\pi(s) = \lim_{t\to\infty} P_{\tau\sim\pi}(s_t = s)$. In the unichain case, the average reward $\rho(\pi)$ does not depend on the initial state for any policy π (Bertsekas et al., 1995). We express the *average-reward bias function* as

$$\bar{V}^\pi(s) := \mathop{\mathbb{E}}_{\tau\sim\pi} \left[\sum_{t=0}^{\infty} (r(s_t, a_t) - \rho(\pi)) \middle| s_0 = s \right]$$

and *average-reward action-bias function* as

$$\bar{Q}^{\pi}(s,a) := \mathop{\mathbb{E}}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}(r(s_t,a_t)-\rho(\pi))\middle| s_0=s, a_0=a\right].$$

We define the *average-reward advantage function* as

$$\bar{A}^{\pi}(s,a) := \bar{Q}^{\pi}(s,a) - \bar{V}^{\pi}(s).$$

Discounted Reward Criterion

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 $d_{\pi,\gamma}(s) := (1-\gamma)\sum_{t=0}^{\infty}\gamma^t P_{\tau\sim\pi}(s_t=s)$ 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 $V_{\gamma}^{\pi}(s) := \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\middle| s_0=s\right]$ and *discounted action-value function* $Q_{\gamma}^{\pi}(s,a) := \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\middle| s_0=s, a_0=a\right]$. Finally, the *discounted advantage function* is defined as $A_{\gamma}^{\pi}(s,a) := Q_{\gamma}^{\pi}(s,a) - V_{\gamma}^{\pi}(s)$.

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 further discuss the relationship between the discounted and average reward criteria in Appendix A and prove that $\lim_{\gamma\to 1} A_{\gamma}^{\pi}(s,a) = \bar{A}^{\pi}(s,a)$ (see Corollary A.1). The proofs of all results in the subsequent sections, if not given, can be found in the supplementary material.

3. Montonically Improvement Guarantees for Discounted RL

In much of the on-policy DRL literature (Schulman et al., 2015; 2017; Wu et al., 2017; Vuong et al., 2019; Song et al., 2020), algorithms iteratively update policies by maximizing them within a local region, i.e., at iteration k we find a policy π_{k+1} by maximizing $\rho_{\gamma}(\pi)$ within some region $D(\pi,\pi_k) \leqslant \delta$ for some divergence measure D. By using different choices of D and δ, this approach allows us to control the step-size of each update, 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:

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

where $D_{\mathrm{TV}}(\pi'\parallel\pi)[s] := \frac{1}{2}\sum_a|\pi'(a|s)-\pi(a|s)|$ is the *total variation divergence*, and $C = 4\gamma\epsilon/(1-\gamma)^2$ where ϵ is some constant. Schulman et al. (2015) showed that by choosing π_{k+1} which maximizes the right hand side of (3), we are guaranteed to have $\rho_{\gamma}(\pi_{k+1}) \geqslant \rho_{\gamma}(\pi_k)$. This provided the theoretical foundation for an entire class of on-policy DRL algorithms (Schulman et al., 2015; 2017; Wu et al., 2017; Vuong et al., 2019; Song et al., 2020).

A natural question arises here is whether the iterative procedure described by Schulman et al. (2015) also guarantees improvement for the average reward. Since the discounted and average reward objectives become equivalent as $\gamma \to 1$, one may conjecture that we can also lower bound the policy performance difference of the average reward objective by simply letting $\gamma \to 1$ for the bounds in Schulman et al. (2015). Unfortunately this results in a non-meaningful bound (see supplementary material for proof.)

Proposition 1. *Consider the bounds in Theorem 1 of Schulman et al. (2015) and Corollary 1 of Achiam et al. (2017). The right hand side of both bounds times* $1-\gamma$ *goes to negative infinity as* $\gamma \to 1$.

Since $\lim_{\gamma\to 1}(1-\gamma)(\rho_{\gamma}(\pi') - \rho_{\gamma}(\pi)) = \rho(\pi') - \rho(\pi)$, Proposition 1 says that the policy improvement guarantee from Schulman et al. (2015) and Achiam et al. (2017) becomes trivial when $\gamma \to 1$ and thus does not generalize to the average reward setting. In the next section, we will derive a novel policy improvement bound for the average reward objective, which in turn 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 $d_{\pi}(s)$. Let $P_{\pi} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ 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 $P_{\pi}^{\star} := \lim_{N\to\infty}\frac{1}{N}\sum_{t=0}^{N}P_{\pi}^t$ be the limiting distribution of the transition matrix. For aperiodic unichain MDPs, $P_{\pi}^{\star} = \lim_{t\to\infty}P_{\pi}^t = \mathbf{1}d_{\pi}^T$.

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 average reward advantange function of π.

Lemma 1. *Under Assumption 2:*

$$\rho(\pi') - \rho(\pi) = \mathop{\mathbb{E}}_{\substack{s\sim d_{\pi'} \\ a\sim\pi'}} \left[\bar{A}^{\pi}(s,a)\right] \tag{4}$$

for any two stochastic policies π and π'.

Lemma 1 is an extension of the well-known policy difference lemma from Kakade & Langford (2002) to the average reward case. A similar result was proven by Even-Dar et al. (2009) and Neu et al. (2010). For completeness, we provide a simple 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" w.r.t. the TV divergence, we can evaluate $\rho(\pi')$ using samples from d_π (see supplementary material for proof).

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

$$\left|\rho(\pi') - \rho(\pi) - \mathop{\mathbb{E}}_{\substack{s\sim d_{\pi} \\ a\sim\pi'}} \left[\bar{A}^{\pi}(s,a)\right]\right| \leqslant 2\epsilon D_{TV}(d_{\pi'} \parallel d_\pi) \tag{5}$$

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

Lemma 2 implies that

$$\rho(\pi') \approx \rho(\pi) + \mathop{\mathbb{E}}_{\substack{s\sim d_{\pi} \\ a\sim\pi'}} \left[\bar{A}^{\pi}(s,a)\right] \tag{6}$$

when d_π and $d_{\pi'}$ are "close". However 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. Under Assumption 1, the matrix M^π can be calculated via (see Theorem 4.4.7 of Kemeny & Snell (1960))

$$M^\pi = (I - Z^\pi + EZ^\pi_{\text{dg}})D^\pi \tag{7}$$

where $Z^\pi = (I - P_\pi + P^\star_\pi)^{-1}$ 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' on some square matrix refers to taking the diagonal of said matrix and placing zeros everywhere else. $D^\pi \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ is a diagonal matrix whose elements are $1/d_\pi(s)$.

One important property of mean first passage time is that for any MDP which satisfies Assumption 1, the quantity

$$\kappa^\pi = \sum_{s'} d_\pi(s') M^\pi(s,s') = \text{trace}(Z^\pi) \tag{8}$$

is a constant independent of the starting state for any policy π (Theorem 4.4.10 of Kemeny & Snell (1960).) 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.

It can be shown that the value of Kemeny's constant is also related to the *mixing time* of the Markov Chain, i.e., how fast the chain converges to the stationary distribution (see Appendix C for additional details).

The following result connects the sensitivity of the stationary distribution to changes to the policy.

Lemma 3. *Under Assumption 1, the divergence between the stationary distributions d_π and $d_{\pi'}$ can be upper bounded by the average divergence between policies π and π':*

$$D_{TV}(d_{\pi'} \parallel d_\pi) \leqslant (\kappa^\star - 1) \mathop{\mathbb{E}}_{s\sim d_\pi} [D_{TV}(\pi' \parallel \pi)[s]] \tag{9}$$

where $\kappa^\star = \max_\pi \kappa^\pi$

For Markov chains with a small mixing time, where an agent can quickly get to any state, Kemeny's constant is relatively small and Lemma 3 shows that the stationary distributions are not highly sensitive to small changes in the policy. On the other hand, for Markov chains that that have high mixing times, the factor can become very large. In this case Lemma 3 shows that small changes in the policy can have a large impact on the resulting stationary distributions.

Combining the bounds in Lemma 2 and Lemma 3 gives us the following result:

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

$$D^-_\pi(\pi') \leqslant \rho(\pi') - \rho(\pi) \leqslant D^+_\pi(\pi') \tag{10}$$

where

$$D^\pm_\pi(\pi') = \mathop{\mathbb{E}}_{\substack{s\sim d_{\pi} \\ a\sim\pi'}} \left[\bar{A}^{\pi}(s,a)\right] \pm 2\xi \mathop{\mathbb{E}}_{s\sim d_\pi} [D_{TV}(\pi' \parallel \pi)[s]]$$

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

The bounds in Theorem 1 are guaranteed to be finite. Analogous to the discounted case, the multiplicative factor ξ provides guidance on the step-sizes for policy updates. Note that Theorem 1 holds for MDPs satisfying Assumption 1; in Appendix D we discuss how a similar result can be derived for the more general aperiodic unichain case.

The bound in Theorem 1 is given in terms of the TV divergence; however the KL divergence is more commonly used in practice. The relationship between the TV divergence and

Algorithm 1 Approximate Average Reward Policy Iteration

Input: π_0
for $k = 0, 1, 2, \ldots$ **do**
  Policy Evaluation: Evaluate $\bar{A}^{\pi_k}(s,a)$ for all s, a
  Policy Improvement:

$$\pi_{k+1} = \operatorname*{argmax}_{\pi} D^{-}_{\pi_k}(\pi) \tag{12}$$

  where

$$D^{-}_{\pi_k}(\pi) = \mathop{\mathbb{E}}_{\substack{s\sim d_{\pi_k}\\ a\sim\pi}} \left[\bar{A}^{\pi_k}(s,a)\right] - \xi\sqrt{2 \mathop{\mathbb{E}}_{s\sim d_{\pi_k}} \left[D_{\mathrm{KL}}\left(\pi \| \pi_k\right)[s]\right]}$$

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

KL divergence is given by Pinsker's inequality (Tsybakov, 2008), which says that for any two distributions p and q: $D_{\mathrm{TV}}(p \parallel q) \leqslant \sqrt{D_{\mathrm{KL}}\left(p\|q\right)/2}$. We can then show that

$$\begin{aligned}\mathop{\mathbb{E}}_{s\sim d_\pi}\left[D_{\mathrm{TV}}(\pi' \parallel \pi)[s]\right] &\leqslant \mathop{\mathbb{E}}_{s\sim d_\pi}\left[\sqrt{D_{\mathrm{KL}}\left(\pi'\|\pi\right)[s]/2}\right] \\ &\leqslant \sqrt{\mathop{\mathbb{E}}_{s\sim d_\pi}\left[D_{\mathrm{KL}}\left(\pi'\|\pi\right)\right][s]/2}\end{aligned} \tag{11}$$

where the second inequality comes from Jensen's inequality. The inequality in (11) shows that the bounds in Theorem 1 still hold when $\mathbb{E}_{s\sim d_\pi}[D_{\mathrm{TV}}(\pi' \parallel \pi)[s]]$ is substituted with $\sqrt{\mathbb{E}_{s\sim d_\pi}[D_{\mathrm{KL}}\left(\pi'\|\pi\right)][s]/2}$.

4.2. Approximate Policy Iteration

One direct consequence of Theorem 1 is that iteratively maximizing the $D^{-}_{\pi}(\pi')$ term in the bound 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 $\rho(\pi_0) \leqslant \rho(\pi_1) \leqslant \rho(\pi_2) \leqslant \cdots$.*

Proof. At iteration k, $\mathbb{E}_{s\sim d_{\pi_k}, a\sim\pi}[\bar{A}^{\pi_k}(s,a)] = 0$, $\mathbb{E}_{s\sim d_{\pi_k}}[D_{\mathrm{KL}}\left(\pi\|\pi_k\right)[s]] = 0$ for $\pi = \pi_k$. By Theorem 1 and (12), $\rho(\pi_{k+1}) - \rho(\pi_k) \geqslant 0$. □

However, Algorithm 1 is difficult to implement in practice since it requires exact knowledge of $\bar{A}^{\pi_k}(s,a)$ and the 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 in Algorithm 1.

5. Practical Algorithm

As noted in the previous section, Algorithm 1 is not practical for problems with large state and action spaces. 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 with the use of *trust region* methods.

In Appendix F, we will also discuss how Theorem 1 can be used to solve DRL problems with average cost safety constraints. RL with safety constraints are an important class of problems with practical implications (Amodei et al., 2016). Trust region methods have been successfully applied to this class of problems as it provides worst-case constraint violation guarantees for evaluating the cost constraint values for policy updates (Achiam et al., 2017; Yang et al., 2020; Zhang et al., 2020). However the aforementioned theoretical guarantees were only shown to apply to discounted cost constraints. Tessler et al. (2019) pointed out that trust-region based methods such as the Constrained Policy Optimization (CPO) algorithm (Achiam et al., 2017) cannot be used for average costs constraints. Contrary to this belief, in Appendix F, we demonstrate that Theorem 1 provides a worst-case constraint violation guarantee for average costs and trust-region-based constrained RL methods can easily be modified to accommodate for average cost constraints.

5.1. Average Reward Trust Region Methods

For DRL problems, it is common to consider some parameterized policy class $\Pi_\Theta = \{\pi_\theta : \theta \in \Theta\}$. Our goal is to devise a computationally tractable version of Algorithm 1 for policies in Π_Θ. We can rewrite the unconstrained optimization problem in (12) as a constrained problem:

$$\begin{aligned}\underset{\pi_\theta\in\Pi_\Theta}{\text{maximize}} \quad & \mathop{\mathbb{E}}_{\substack{s\sim d_{\pi_{\theta_k}}\\ a\sim\pi_\theta}} \left[\bar{A}^{\pi_{\theta_k}}(s,a)\right] \\ \text{subject to} \quad & \bar{D}_{\mathrm{KL}}(\pi_\theta \parallel \pi_{\theta_k}) \leqslant \delta\end{aligned} \tag{13}$$

where $\bar{D}_{\mathrm{KL}}(\pi_\theta \parallel \pi_{\theta_k}) := \mathbb{E}_{s\sim d_{\pi_{\theta_k}}}\left[D_{\mathrm{KL}}\left(\pi_\theta\|\pi_{\theta_k}\right)[s]\right]$. Importantly, the advantage function $\bar{A}^{\pi_{\theta_k}}(s,a)$ appearing in (13) is the average-reward advantage function, defined as the bias minus the action-bias, and not the discounted advantage function. The constraint set $\{\pi_\theta \in \Pi_\Theta : \bar{D}_{\mathrm{KL}}(\pi_\theta \parallel \pi_{\theta_k}) \leqslant \delta\}$ is called the *trust region set*. The problem (13) can be regarded as an average reward variant of the trust region problem from Schulman et al. (2015). The step-size δ is treated as a hyperparamter in practice and should ideally be tuned for each specific task. However we note that in the average reward, the choice of step-size is related to the mixing time of the underlying Markov chain (since it is related to the multiplicative factor ξ in Theorem 1). When the mixing time is small, a larger step-size can be chosen and vice versa. While it is impractical to calculate the optimal

step-size, in certain applications domain knowledge on the mixing time can be used to serve as a guide for tuning δ.

When we set $\pi_{\theta_{k+1}}$ to be the optimal solution to (13), similar to the discounted case, the policy improvement guarantee no longer holds. However we can show that $\pi_{\theta_{k+1}}$ has the following worst-case performance degradation guarantee:

Proposition 3. *Let $\pi_{\theta_{k+1}}$ be the optimal solution to (13) 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*

$$\rho(\pi_{\theta_{k+1}}) - \rho(\pi_{\theta_k}) \geqslant -\xi^{\pi_{\theta_{k+1}}} \sqrt{2\delta} \tag{14}$$

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

Proof. Since $\bar{D}_{\mathrm{KL}}(\pi_{\theta_k} \,\|\, \pi_{\theta_k}) = 0$, π_{θ_k} is feasible. The objective value is 0 for $\pi_\theta = \pi_{\theta_k}$. The bound follows from (10) and (11) where the average KL is bounded by δ. □

Several algorithms have been proposed for efficiently solving the discounted version of (13): Schulman et al. (2015) and Wu et al. (2017) converts (13) into a convex problem via Taylor approximations; another approach is to first solve (13) in the non-parametric policy space and then project the result back into the parameter space (Vuong et al., 2019; Song et al., 2020). These algorithms can also be adapted for the average reward case and are theoretically justified via Theorem 1 and Proposition 3. In the next section, we will provide as a specific example how this can be done for one such algorithm.

5.2. Average Reward TRPO (ATRPO)

In this section, we introduce ATRPO, which is an average-reward modification of the TRPO algorithm (Schulman et al., 2015). Similar to TRPO, we apply Taylor approximations to (13). This gives us a new optimization problem which can be solved exactly using Lagrange duality (Boyd et al., 2004). The solution to this approximate problem gives an explicit update rule for the policy parameters which then allows us to perform policy updates using an actor-critic framework. More details can be found in Appendix E. Algorithm 2 provides a basic outline of ATRPO.

The major differences between ATRPO and TRPO are as follows:

i The critic network in Algorithm 2 approximates the average-reward bias rather than the discounted value function.

ii ATRPO must estimate the average return ρ of the current policy.

iii The targets for the bias and the advantage are calculated without discount factors and the average return ρ is subtracted from the reward. Simply setting the discount factor to 1 in TRPO does not lead to Algorithm 2.

iv ATRPO also assumes that the underlying task is a continuing infinite-horizon task. But since in practice we cannot run infinitely long trajectories, all trajectories are truncated at some large truncation value N. Unlike TRPO, during training we do not allow for episodic tasks where episodes terminate early (before N). For the MuJoCo environments, we will address this by having the agent not only resume locomotion after falling but also incur a penalty for falling (see Section 6.)

Algorithm 2 Average Reward TRPO (ATRPO)

Input: Policy parameters θ_0, critic net parameters ϕ_0, learning rate α, trajectory truncation parameter N.
for $k = 0, 1, 2, \cdots$ **do**
  Collect a truncated trajectory $\{s_t, a_t, s_{t+1}, r_t\}$, $t = 1, \ldots, N$ from the environment using π_{θ_k}.
  Calculate sample average reward of π_{θ_k} via $\rho = \frac{1}{N}\sum_{t=1}^{N} r_t$.
  for $t = 1, 2, \ldots, N$ **do**
    Get target $\bar{V}_t^{\text{target}} = r_t - \rho + \bar{V}_{\phi_k}(s_{t+1})$
    Get advantage estimate: $\hat{A}(s_t, a_t) = r_t - \rho + \bar{V}_{\phi_k}(s_{t+1}) - \bar{V}_{\phi_k}(s_t)$
  end for
  Update critic by

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

  where

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

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

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 develop and use an average-reward version of the Generalized Advantage Estimator (GAE) from Schulman et al. (2016). In Appendix G we provide more details on how GAE can be generalized to the average-reward case.

6. Experiments

We conducted experiments comparing the performance of ATRPO and TRPO on continuing control tasks. We consider three tasks (Ant, HalfCheetah, and Humanoid) from the MuJoCo physical simulator (Todorov et al., 2012) imple-

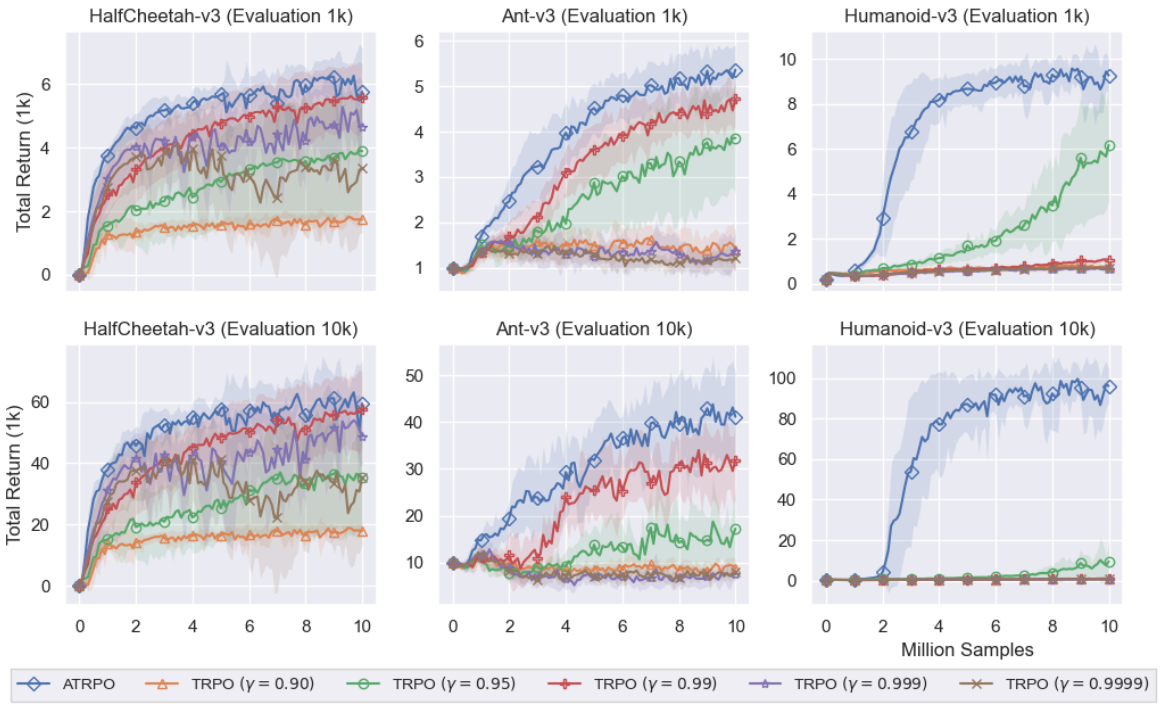

Figure 1. Comparing performance of ATRPO and TRPO with different discount factors. The x-axis is the number of agent-environment interactions and the y-axis is the total return averaged over 10 seeds. The solid line represents the agents' performance on evaluation trajectories of maximum length 1,000 (top row) and 10,000 (bottom row). The shaded region represents one standard deviation.

mented using OpenAI gym (Brockman et al., 2016), where the natural goal is to train the agents to run as fast as possible without falling.

6.1. Evaluation Protocol

Even though the MuJoCo benchmark is commonly trained using the *discounted* objective (see e.g. Schulman et al. (2015), Wu et al. (2017), Lillicrap et al. (2016), Schulman et al. (2017), Haarnoja et al. (2018), Vuong et al. (2019)), it is *always* evaluated without discounting. Similarly, we also evaluate performance using the undiscounted total-reward objective for both TRPO and ATRPO.

Specifically for each environment, we train a policy for 10 million environment steps. During training, every 100,000 steps, we run 10 separate evaluation trajectories with the current policy without exploration (i.e., the policy is kept fixed and deterministic). For each evaluation trajectory we calculate the undiscounted return of the trajectory until the agent falls or until 1,000 steps, whichever comes first. We then report the average undiscounted return over the 10 trajectories. *Note that this is the standard evaluation metric for the MuJoCo environments.* In order to understand the performance of the agent for long time horizons, we also report the performance of the agent evaluated on trajectories of maximum length 10,000.

6.2. Comparing ATRPO and TRPO

To simulate an infinite-horizon setting during training, we do the following: when the agent falls, the trajectory does not terminate; instead the agent incurs a large reset cost for falling, and then continues the trajectory from a random start state. The reset cost is set to 100. However, we show in the supplementary material (Appendix I.2) that the results are largely insensitive to the choice of reset cost. We note that this modification does not change the underlying goal of the task. We also point out that the reset cost is only applied during training and is not used in the evaluation phase described in the previous section. Hyperparameter settings and other additional details can be found in Appendix H.

We plot the performance for ATRPO and TRPO trained with different discount factors in Figure 1. We see that TRPO with its best discount factor can perform as well as ATRPO for the simplest environment HalfCheetah. But ATRPO provides dramatic improvements in Ant and Humanoid. In particular for the most challenging environment Humanoid, ATRPO performs on average 50.1% better than TRPO with its best discount factor when evaluated on trajectories of maximum length 1000. The improvement is even greater when the agents are evaluated on trajectories of maximum length 10,000 where the performance boost jumps to 913%. In Appendix I.1, we provide an additional set of experiments

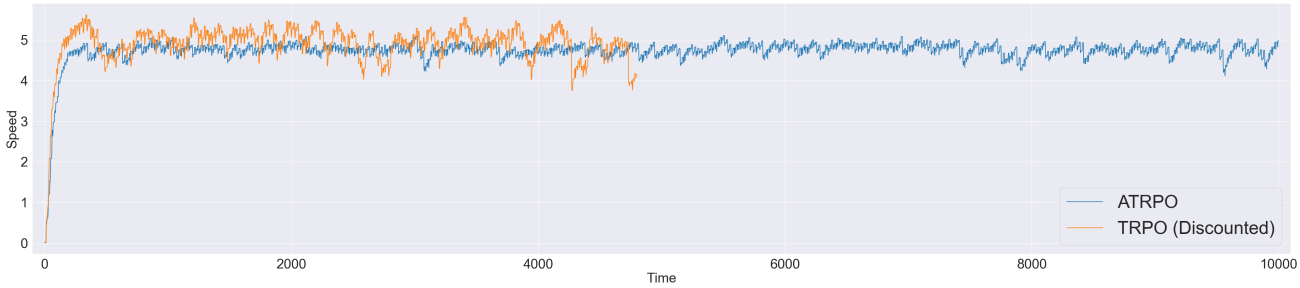

Figure 2. Speed-time plot of a single trajectory (maximum length 10,000) for ATRPO and Discounted TRPO in the Humanoid-v3 environment. The solid line represents the speed of the agent at the corresponding timesteps.

demonstrating that ATRPO also significantly outperforms TRPO when TRPO is trained without the reset scheme described at the beginning of this section (i.e. the standard MuJoCo setting.)

We make two observations regarding discounting. First, we note that increasing the discount factor does not necessarily lead to better performance for TRPO. A larger discount factor in principle enables the algorithm to seek a policy that performs well for the average-reward criterion (Blackwell, 1962). Unfortunately, a larger discount factor can also increase the variance of the gradient estimator (Zhao et al., 2011; Schulman et al., 2016), increase the complexity of the policy space (Jiang et al., 2015), lead to slower convergence (Bertsekas et al., 1995; Agarwal et al., 2020), and degrade generalization in limited data settings (Amit et al., 2020). Moreover, algorithms with discounting are known to become unstable as $\gamma \to 1$ (Naik et al., 2019). Secondly, for TRPO the best discount factor is different for each environment (0.99 for HalfCheetah and Ant, 0.95 for Humanoid). The discount factor therefore serves as a hyperparameter which can be tuned to improve performance, choosing a suboptimal discount factor can have significant consequences. Both of these observation are consistent with what was seen in the literature (Andrychowicz et al., 2020). We have shown here that using the average reward criterion directly not only delivers superior performance but also obviates the need to tune the discount factor.

6.3. Understanding Long Run Performance

Next, we demonstrate that agents trained using the average reward criterion are better at optimizing for long-term returns. Here, we first train Humanoid with 10 million samples with ATRPO and with TRPO with a discount factor of 0.95 (shown to be the best discount factor in the previous experiments). Then for evaluation, we run the trained ATRPO and TRPO policies for a trajectory of 10,000 timesteps (or until the agent falls). We use the same random seeds for the two algorithms. Figure 2 is a plot of the speed of the agent at each time step of the trajectory, using the *seed that gives the best performance for discounted TRPO*. We see in Figure 2 that the discounted algorithm gives a higher initial speed at the beginning of the trajectory. However its overall speed is much more erratic throughout the trajectory, resulting in the agent falling over after approximately 5000 steps. This coincides with the notion of discounting where more emphasis is placed at the beginning of the trajectory and ignores longer-term behavior. On the other hand, the average-reward policy — while having a slightly lower velocity overall throughout its trajectory — is able to sustain the trajectory much longer, thus giving it a higher total return. In fact, we observed that for all 10 random seeds we tested, the average reward agent is able to finish the entire 10,000 time step trajectory without falling. In Table 1 we present the summary statistics of trajectory length for all trajectories using discounted TRPO we note that the median trajectory length for the TRPO discounted agent is 452.5, meaning that on average TRPO performs significantly worse than what is reported in Figure. 2.

Table 1. Summary statistics for all 10 trajectories using a Humanoid-v3 agent trained with TRPO

Min	Max	Average	Median	Std
108	4806	883.1	452.5	1329.902

7. Related Work

Dynamic programming algorithms for finding the optimal average reward policies have been well-studied (Howard, 1960; Blackwell, 1962; Veinott, 1966). Several tabular 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), CSV-Learning (Yang et al., 2016), and Differential Q-Learning (Wan et al., 2020). 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.

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 (2001a) 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 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) are extensions of Kakade & Langford (2002). 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.

8. Conclusion

In this paper, we introduce 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 or mixing time of the Markov chain. We show that previous existing policy improvement bounds for the discounted case results in a non-meaningful bound for the average reward objective. Our work provides the theoretical justification and the means to generalize the popular trust-region based algorithms to the average reward setting. Based on this theory, we propose ATRPO, a modification of the TRPO algorithm for on-policy DRL. We demonstrate through a series of experiments that ATRPO is highly effective on high-dimensional continuing control tasks.

Acknowledgements

We would like to extend our gratitude to Quan Vuong and the anonymous reviewers for their constructive comments and suggestions. We also thank Shuyang Ling, Che Wang, Zining (Lily) Wang, and Yanqiu Wu for the insightful discussions on this work.

References

Abbasi-Yadkori, Y., Bartlett, P., Bhatia, K., Lazic, N., Szepesvari, C., and Weisz, G. Politex: Regret bounds for policy iteration using expert prediction. In *International Conference on Machine Learning*, pp. 3692–3702, 2019.

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

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

Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. Optimality and approximation with policy gradient methods in markov decision processes. In *Conference on Learning Theory*, pp. 64–66. PMLR, 2020.

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

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

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Andrychowicz, M., Raichuk, A., Stańczyk, P., Orsini, M., Girgin, S., Marinier, R., Hussenot, L., Geist, M., Pietquin, O., Michalski, M., et al. What matters in on-policy reinforcement learning? a large-scale empirical study. *arXiv preprint arXiv:2006.05990*, 2020.

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

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

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

Boyd, S., Boyd, S. P., and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

Brémaud, P. *Markov Chains Gibbs Fields, Monte Carlo Simulation and Queues*. Springer, 2 edition, 2020.

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

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

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

Grinstead, C. M. and Snell, J. L. *Introduction to probability*. American Mathematical Soc., 2012.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *International Conference on Machine Learning (ICML)*, 2018.

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

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

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

Jiang, N., Kulesza, A., Singh, S., and Lewis, R. The dependence of effective planning horizon on model accuracy. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pp. 1181–1189. Citeseer, 2015.

Jiang, N., Singh, S. P., and Tewari, A. On structural properties of mdps that bound loss due to shallow planning. In *IJCAI*, pp. 1640–1647, 2016.

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

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

Kakade, S. M. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001b.

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

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

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

Lehnert, L., Laroche, R., and van Seijen, H. On value function representation of long horizon problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Levin, D. A. and Peres, Y. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

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

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

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

Naik, A., Shariff, R., Yasui, N., and Sutton, R. S. Discounted reinforcement learning is not an optimization problem. *NeurIPS Optimization Foundations for Reinforcement Learning Workshop*, 2019.

Neu, G., Antos, A., György, A., and Szepesvári, C. Online markov decision processes under bandit feedback. In *Advances in Neural Information Processing Systems*, pp. 1804–1812, 2010.

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

Petrik, M. and Scherrer, B. Biasing approximate dynamic programming with a lower discount factor. In *Twenty-Second Annual Conference on Neural Information Processing Systems-NIPS 2008*, 2008.

Pirotta, M., Restelli, M., Pecorino, A., and Calandriello, D. Safe policy iteration. In *International Conference on Machine Learning*, pp. 307–315, 2013.

Ross, K. W. 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.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International Conference on Machine Learning*, pp. 1889–1897, 2015.

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

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

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

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

Song, H. F., Abdolmaleki, A., Springenberg, J. T., Clark, A., Soyer, H., Rae, J. W., Noury, S., Ahuja, A., Liu, S., Tirumala, D., et al. V-mpo: on-policy maximum a posteriori policy optimization for discrete and continuous control. *International Conference on Learning Representations*, 2020.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pp. 1057–1063, 2000.

Tadepalli, P. and Ok, D. H-learning: A reinforcement learning method to optimize undiscounted average reward. Technical Report 94-30-01, Oregon State University, 1994.

Tessler, C., Mankowitz, D. J., and Mannor, S. Reward constrained policy optimization. *International Conference on Learning Representation (ICLR)*, 2019.

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

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

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

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

Wan, Y., Naik, A., and Sutton, R. S. Learning and planning in average-reward markov decision processes. *arXiv preprint arXiv:2006.16318*, 2020.

Wei, C.-Y., Jafarnia-Jahromi, M., Luo, H., Sharma, H., and Jain, R. Model-free reinforcement learning in infinite-horizon average-reward markov decision processes. In *International conference on machine learning*, 2020.

Wu, Y., Mansimov, E., Grosse, R. B., Liao, S., and Ba, J. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. In *Advances in neural information processing systems (NIPS)*, pp. 5285–5294, 2017.

Yang, S., Gao, Y., An, B., Wang, H., and Chen, X. Efficient average reward reinforcement learning using constant shifting values. In *AAAI*, pp. 2258–2264, 2016.

Yang, T.-Y., Rosca, J., Narasimhan, K., and Ramadge, P. J. Projection-based constrained policy optimization. In *International Conference on Learning Representation (ICLR)*, 2020.

Zhang, Y., Vuong, Q., and Ross, K. First order constrained optimization in policy space. *Advances in Neural Information Processing Systems*, 33, 2020.

Zhao, T., Hachiya, H., Niu, G., and Sugiyama, M. Analysis and improvement of policy gradient estimation. In *Advances in Neural Information Processing Systems*, pp. 262–270, 2011.

On-Policy Deep Reinforcement Learning for the Average-Reward Criterion Supplementary Materials

A. Relationship Between the Discounted and Average Reward Criteria

We first introduce the average reward Bellman equations (Sutton & Barto, 2018):

$$\bar{V}^\pi(s) = \sum_a \pi(a|s) \left[r(s,a) - \rho(\pi) + \sum_{s'} P(s'|s,a) \bar{V}^\pi(s') \right] \tag{15}$$

$$\bar{Q}^\pi(s,a) = r(s,a) - \rho(\pi) + \sum_{s'} P(s'|s,a) \sum_{a'} \pi(a'|s') \bar{Q}^\pi(s',a'). \tag{16}$$

From which we can easily show that:

$$\bar{V}^\pi(s) = \sum_a \pi(a|s) \bar{Q}^\pi(s,a) \tag{17}$$

$$\bar{Q}^\pi(s,a) = r(s,a) - \rho(\pi) + \sum_{s'} P(s'|s,a) \bar{V}^\pi(s'). \tag{18}$$

Note that these equations take a slightly different form compared to the discounted Bellman equations, there are no discount factors and the rewards are now replaced with $r(s,a) - \rho(\pi)$.

The following classic result relates the value function in the discounted case and average reward bias functions.

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

$$\lim_{\gamma \to 1} \left(V_\gamma^\pi(s) - \frac{\rho(\pi)}{1-\gamma} \right) = \bar{V}^\pi(s) \tag{19}$$

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

Note that Proposition A.1 applies to any MDP, we will however restrict our discussion to the unichain case to coincide with the scope of the paper. 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-bias function and advantage function.

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

$$\lim_{\gamma \to 1} \left(Q_\gamma^\pi(s,a) - \frac{\rho(\pi)}{1-\gamma} \right) = \bar{Q}^\pi(s,a) \tag{20}$$

$$\lim_{\gamma \to 1} A_\gamma^\pi(s,a) = \bar{A}^\pi(s,a) \tag{21}$$

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

Proof. From Proposition A.1, we can rewrite (19) as

$$V_\gamma^\pi(s) = \frac{\rho(\pi)}{1-\gamma} + \bar{V}^\pi(s) + g(\gamma, s) \tag{22}$$

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

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

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 (20).

We can then similarly rewrite (20) as

$$Q^\pi_\gamma(s,a) = \frac{\rho(\pi)}{1-\gamma} + \bar{Q}^\pi(s,a) + h(\gamma, s, a) \tag{23}$$

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

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

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 (21). □

B. Proofs

B.1. Proof of Proposition 1

Proposition 1. *Consider the bounds in Theorem 1 of Schulman et al. (2015) and Corollary 1 of Achiam et al. (2017). The right hand side of both bounds times $1-\gamma$ goes to negative infinity as $\gamma \to 1$.*

Proof. We will give a proof for the case of Corollary 1 in Achiam et al. (2017), a similar argument can be applied to the bound in Theorem 1 of Schulman et al. (2015).

We first state Corollary 1 of Achiam et al. (2017) which says that for any two stationary policies π and π':

$$\rho_\gamma(\pi') - \rho_\gamma(\pi) \geqslant \frac{1}{1-\gamma} \left[\mathop{\mathbb{E}}_{\substack{s \sim d_{\pi,\gamma} \\ a \sim \pi'}} [A^\pi_\gamma(s,a)] - \frac{2\gamma\epsilon^\gamma}{1-\gamma} \mathop{\mathbb{E}}_{s \sim d_{\pi,\gamma}} D_{\text{TV}}(\pi' \parallel \pi) \right] \tag{24}$$

where $\epsilon^\gamma = \max_s \left| \mathbb{E}_{a \sim \pi'} [A^\pi_\gamma(s,a)] \right|$. Since $d_{\pi,\gamma}$ approaches the stationary distribution d_π as $\gamma \to 1$, we can multiply the

right hand side of (24) by $(1-\gamma)$ and take the limit which gives us:

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

Here $\epsilon = \max_s \left|\mathbb{E}_{a\sim\pi'}[\bar{A}^{\pi}(s,a)]\right|$. The first equality is a direct result of Corollary A.1. □

B.2. Proof of Lemma 1

Lemma 1. *Under Assumption 2:*

$$\rho(\pi') - \rho(\pi) = \mathop{\mathbb{E}}_{\substack{s\sim d_{\pi'}\\ a\sim\pi'}}\left[\bar{A}^{\pi}(s,a)\right] \tag{4}$$

for any two stochastic policies π and π'.

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

$$\begin{aligned}
\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi'}\\ a\sim\pi'}}\left[\bar{A}^{\pi}(s,a)\right] &= \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi'}\\ a\sim\pi'}}\left[\bar{Q}^{\pi}(s,a) - \bar{V}^{\pi}(s)\right]\\
&= \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi'}\\ a\sim\pi'}}\left[r(s,a) - \rho(\pi) + \mathop{\mathbb{E}}_{s'\sim P(\cdot|s,a)}\left[\bar{V}^{\pi}(s')\right] - \bar{V}^{\pi}(s)\right]\\
&= \rho(\pi') - \rho(\pi) + \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi'}\\ a\sim\pi'\\ s'\sim P(\cdot|s,a)}}[\bar{V}^{\pi}(s')] - \mathop{\mathbb{E}}_{s\sim d^{\pi'}}[\bar{V}^{\pi}(s)]
\end{aligned}$$

Since $d_{\pi'}(s)$ is the stationary distribution:

$$\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi'}\\ a\sim\pi'\\ s'\sim P(\cdot|s,a)}}[\bar{V}^{\pi}(s')] = \sum_s d_{\pi'}(s)\sum_a \pi'(a|s)\sum_{s'} P(s'|s,a)\bar{V}^{\pi}(s') = \sum_s d_{\pi'}(s)\sum_{s'} P_{\pi'}(s'|s)\bar{V}^{\pi}(s') = \sum_{s'} d_{\pi'}(s')\bar{V}^{\pi}(s')$$

Therefore,

$$\mathop{\mathbb{E}}_{\substack{s\sim d^{\pi'}\\ a\sim\pi'\\ s'\sim P(\cdot|s,a)}}[\bar{V}^{\pi}(s')] - \mathop{\mathbb{E}}_{s\sim d^{\pi'}}[\bar{V}^{\pi}(s)] = 0$$

which gives us the desired result.

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. *Under Assumption 2, the following bound holds for any two stochastic policies π and π':*

$$\left|\rho(\pi') - \rho(\pi) - \mathop{\mathbb{E}}_{\substack{s\sim d_{\pi}\\ a\sim\pi'}}\left[\bar{A}^{\pi}(s,a)\right]\right| \leqslant 2\epsilon D_{TV}(d_{\pi'} \parallel d_{\pi}) \tag{5}$$

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

Proof.

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

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

B.4. Proof of Lemma 3

Lemma 3. *Under Assumption 1, the divergence between the stationary distributions d_π and $d_{\pi'}$ can be upper bounded by the average divergence between policies π and π':*

$$D_{TV}(d_{\pi'} \parallel d_\pi) \leqslant (\kappa^\star - 1) \mathop{\mathbb{E}}_{s\sim d_\pi}[D_{TV}(\pi' \parallel \pi)[s]] \tag{9}$$

where $\kappa^\star = \max_\pi \kappa^\pi$

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

$$\begin{aligned}
(d_{\pi'}^T - d_\pi^T)(I - P_{\pi'} + P_{\pi'}^\star) &= d_{\pi'}^T - d_\pi^T - d_{\pi'}^T + d_\pi^T P_{\pi'} \\
&= d_\pi^T P_{\pi'} - d_\pi^T \\
&= d_\pi^T (P_{\pi'} - P_\pi)
\end{aligned} \tag{25}$$

Right multiplying (25) by $(I - P_{\pi'} + P_{\pi'}^\star)^{-1}$ gives us:

$$d_{\pi'}^T - d_\pi^T = d_\pi^T (P_{\pi'} - P_\pi)(I - P_{\pi'} + P_{\pi'}^\star)^{-1} \tag{26}$$

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

$$Z^{\pi'} = I + EZ_{\text{dg}}^{\pi'} - M^{\pi'}(D^{\pi'})^{-1} \tag{27}$$

Plugging (27) into (26) gives us

$$\begin{aligned}
d_{\pi'}^T - d_\pi^T &= d_\pi^T (P_{\pi'} - P_\pi)(I + EZ_{\text{dg}}^{\pi'} - M^{\pi'}(D^{\pi'})^{-1}) \\
&= d_\pi^T (P_{\pi'} - P_\pi)(I - M^{\pi'}(D^{\pi'})^{-1})
\end{aligned} \tag{28}$$

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

Let $\|\cdot\|_p$ denote the the operator norm of a matrix, in particular $\|\cdot\|_1$ and $\|\cdot\|_\infty$ are the maximum absolute column sum and maximum absolute row sum respectively. By the submultiplicative property of operator norms (Horn & Johnson, 2012), we have:

$$\begin{aligned}
\|d_{\pi'} - d_\pi\|_1 &= \left\|(I - M^{\pi'}(D^{\pi'})^{-1})^T (P_{\pi'}^T - P_\pi^T) d_\pi\right\|_1 \\
&\leqslant \left\|(I - M^{\pi'}(D^{\pi'})^{-1})^T\right\|_1 \left\|(P_{\pi'}^T - P_\pi^T) d_\pi\right\|_1 \\
&= \left\|(I - M^{\pi'}(D^{\pi'})^{-1})\right\|_\infty \left\|(P_{\pi'}^T - P_\pi^T) d_\pi\right\|_1
\end{aligned} \tag{29}$$

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

$$\begin{aligned}\left\|I - M^{\pi'}(D^{\pi'})^{-1}\right\|_\infty &= \max_s \left(\sum_{s'} M^{\pi'}(s,s')d_{\pi'}(s') - 1\right) \\ &= \kappa^{\pi'} - 1\end{aligned} \tag{30}$$

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

$$\begin{aligned}\left\|(P_{\pi'}^T - P_\pi^T)d_\pi\right\|_1 &= \sum_{s'}\left|\sum_s \left(\sum_a P(s'|s,a)\pi'(a|s) - P(s'|s,a)\pi(a|s)\right) d_\pi(s)\right| \\ &\leqslant \sum_{s',s}\left|\sum_a P(s'|s,a)(\pi'(a|s) - \pi(a|s))\right| d_\pi(s) \\ &\leqslant \sum_{s,s',a} P(s'|s,a)\left|\pi'(a|s) - \pi(a|s)\right| d_\pi(s) \\ &\leqslant \sum_{s,a}\left|\pi'(a|s) - \pi(a|s)\right| d_\pi(s) \\ &= 2 \mathop{\mathbb{E}}_{s\sim d^\pi}[D_{\text{TV}}(\pi' \parallel \pi)]\end{aligned} \tag{31}$$

Plugging back into (29) and setting $\kappa^\star = \max_\pi \kappa^\pi$ gives the desired result. □

C. Kemeny's Constant and Mixing Time

Proposition C.1. *Under Assumption 1, let* $1 = \lambda_1(\pi) > \lambda_2(\pi) \geqslant \cdots \geqslant \lambda_{|\mathcal{S}|}(\pi) > -1$ *be the eigenvalues of* P_π*, we have*

$$\kappa^\pi \leqslant 1 + \frac{|\mathcal{S}| - 1}{1 - \lambda^\star(\pi)} \tag{32}$$

where $\lambda^\star(\pi) = \max_{i=2,\ldots,|\mathcal{S}|} |\lambda_i(\pi)|$.

Proof. For brevity, we omit π from the notations in our proof. Let λ be an eigenvalue of P and u its corresponding eigenvector. Since P is aperiodic, $\lambda \neq -1$, we then have

$$\begin{aligned}(I - P + P^\star)u &= u - Pu + \lim_{n\to\infty} P^n u \\ &= (1-\lambda)u + u \lim_{n\to\infty} \lambda^n \\ &= \left(1 - \lambda + \lim_{n\to\infty} \lambda^n\right) u\end{aligned} \tag{33}$$

where $\lim_{n\to\infty} \lambda^n = 1$ when $\lambda = 1$ and 0 when $|\lambda| < 1$. Therefore, $(I - P + P^\star)$ has eigenvalues $1, 1-\lambda_2, \ldots, 1-\lambda_{|\mathcal{S}|}$. The fundamental matrix $Z = (I - P + P^\star)^{-1}$ has eigenvalues $1, \frac{1}{1-\lambda_2}, \cdots, \frac{1}{1-\lambda_{|\mathcal{S}|}}$. We can then upper bound Kemeny's constant by

$$\kappa = \text{trace}(Z) = 1 + \sum_{i=2}^{|\mathcal{S}|} \frac{1}{1-\lambda_i} \leqslant 1 + \sum_{i=2}^{|\mathcal{S}|} \frac{1}{1-|\lambda_i|} \leqslant 1 + \frac{|\mathcal{S}| - 1}{1 - \lambda^\star} \tag{34}$$

□

The expression $\lambda^\star(\pi)$ is called as the *Second Largest Eigenvalue Modulo (SLEM)*. The Perron-Frobenius theorem says that the transition matrix P_π converges to the limiting distribution $P_\pi^\star$ at an exponential rate, and the rate of convergence is determined by the SLEM (see Theorem 4.3.8 of Brémaud (2020) for more details). In fact, it turns out that the mixing time of a Markov chain is directly related to the SLEM where Markov chains with larger SLEM takes longer to mix and vice versa (Levin & Peres, 2017).

D. Average Reward Policy Improvement Bound for Aperiodic Unichain MDPs

In this section, we consider general aperiodic unichain MDPs, i.e. MPDs which satisfy Assumption 2.

We note that Lemma 1 and Lemma 2 both hold under Assumption 2. We can then show the following under the general aperiodic unichain case:

Lemma D.1. *For any aperiodic unichain MDP:*

$$D_{TV}(d_{\pi'} \parallel d_\pi) \leqslant \zeta^\star \mathop{\mathbb{E}}_{s\sim d_\pi} [D_{TV}(\pi' \parallel \pi)[s]] \tag{35}$$

where $\zeta^\star = \max_\pi \|Z^\pi\|_\infty$.

Proof. Note that

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

from Equation 26 still holds in the general aperiodic unichain case. By the submultiplicative property, we have:

$$\begin{aligned} \|d_{\pi'} - d_\pi\|_1 &= \left\|((I - P_{\pi'} + P_{\pi'}^\star)^{-1})^T (P_{\pi'}^T - P_\pi^T) d_\pi\right\|_1 \\ &\leqslant \left\|((I - P_{\pi'} + P_{\pi'}^\star)^{-1})^T\right\|_1 \left\|(P_{\pi'}^T - P_\pi^T) d_\pi\right\|_1 \\ &= \left\|(I - P_{\pi'} + P_{\pi'}^\star)^{-1}\right\|_\infty \left\|(P_{\pi'}^T - P_\pi^T) d_\pi\right\|_1 \end{aligned} \tag{36}$$

Using the same argument as (31) to bound $\left\|(P_{\pi'}^T - P_\pi^T) d_\pi\right\|_1$ and setting $\zeta^\star = \max_\pi \|Z^\pi\|_\infty$ gives the desired result. □

Combining Lemma 2 and Lemma D.1 gives us the following result:

Theorem 2. *For any aperiodic unichain MDP, the following bounds hold for any two stochastic policies* π *and* π'*:*

$$\rho(\pi') - \rho(\pi) \leqslant \mathop{\mathbb{E}}_{\substack{s\sim d_\pi \\ a\sim\pi'}} \left[\bar{A}^\pi(s,a)\right] + 2\tilde{\xi} \mathop{\mathbb{E}}_{s\sim d_\pi} [D_{TV}(\pi' \parallel \pi)[s]] \tag{37}$$

$$\rho(\pi') - \rho(\pi) \geqslant \mathop{\mathbb{E}}_{\substack{s\sim d_\pi \\ a\sim\pi'}} \left[\bar{A}^\pi(s,a)\right] - 2\tilde{\xi} \mathop{\mathbb{E}}_{s\sim d_\pi} [D_{TV}(\pi' \parallel \pi)[s]] \tag{38}$$

where $\tilde{\xi} = \zeta^\star \max_s \mathbb{E}_{a\sim\pi'} |\bar{A}^\pi(s,a)|$.

The constant $\zeta^\star$ is always finite therefore we can similarly apply the approximate policy iteration procedure from Algorithm 1 to generate a sequence of monotonically improving policies.

E. Derivation of ATRPO

In this section, we give the derivation and additional details of the ATRPO algorithm presented in Algorithm 2. The algorithm is similar to TRPO in the discount case but with several notable distinctions. Recall the trust region optimization problem from (13) where

$$\begin{aligned} \underset{\pi_\theta\in\Pi_\theta}{\text{maximize}} \quad & \mathop{\mathbb{E}}_{\substack{s\sim d_{\pi_{\theta_k}} \\ a\sim\pi_\theta}} [\bar{A}^{\pi_{\theta_k}}(s,a)] \\ \text{subject to} \quad & \bar{D}_{\text{KL}}(\pi_\theta \parallel \pi_{\theta_k}) \leqslant \delta \end{aligned}$$

We once again note that the objective above is the expectation of the *average-reward* advantage function and not the standard discounted advantage function. As done for the derivation for discounted TRPO, we can approximate this problem by performing first-order Taylor approximation on the objective and second-order approximation on the KL constraint[1] around θ_k which gives us:

$$\begin{aligned} \underset{\theta}{\text{maximize}} \quad & g^T(\theta - \theta_k) \\ \text{subject to} \quad & \frac{1}{2}(\theta - \theta_k)^T H (\theta - \theta_k) \leqslant \delta \end{aligned} \tag{39}$$

[1]The gradient and first-order Taylor approximation of $\bar{D}_{\text{KL}}(\pi_\theta \parallel \pi_{\theta_k})$ at $\theta = \theta_k$ is zero.

where

$$g := \mathop{\mathbb{E}}_{\substack{s\sim d_{\pi_{\theta_k}} \\ a\sim\pi_{\theta_k}}} \left[\nabla_\theta \log \pi_\theta(a|s)|_{\theta=\theta_k} \bar{A}^{\pi_{\theta_k}}(s,a)\right] \tag{40}$$

and

$$H := \mathop{\mathbb{E}}_{\substack{s\sim d_{\pi_{\theta_k}} \\ a\sim\pi_{\theta_k}}} \left[\nabla_\theta \log \pi_\theta(a|s)|_{\theta=\theta_k} \nabla_\theta \log \pi_\theta(a|s)|^T_{\theta=\theta_k}\right] \tag{41}$$

Note that this approximation is good provided that the step-size δ is small. The term g is the average reward policy gradient at $\theta = \theta_k$ with an additional baseline term (Sutton et al., 2000) and H is the *Fisher Information Matrix* (FIM) (Lehmann & Casella, 2006). The FIM is a symmetrical matrix and always positive semi-definite. If we assume H is always positive definite, we can solve (39) analytically with a Lagrange duality argument which yields the solution:

$$\theta = \theta_k + \sqrt{\frac{2\delta}{g^T H^{-1} g}} H^{-1} g \tag{42}$$

The update rule in (42) has the same form as that of natural policy gradients (Kakade, 2001b) for the average reward case. Similar to discounted TRPO, both g and H can be approximated using samples drawn from the policy π_{θ_k}. The FIM H here is identical to the FIM H for Natural Gradient and TRPO. However, the definition of g is different from the definition of g for discounted TRPO since it includes the average-reward advantage function.

Thus, in order to estimate g we need to estimate

$$\bar{A}^{\pi_{\theta_k}}(s,a) = \bar{Q}^{\pi_{\theta_k}}(s,a) - \bar{V}^{\pi_{\theta_k}}(s) \tag{43}$$

This can be done in various ways. One approach is to approximate the average-reward bias $\bar{V}^{\pi_{\theta_k}}(s)$ and then use a one-step TD backup (as was done in Algorithm 2) to estimate the action-bias function. Concretely, combining (43) and the Bellman equation in (18) gives

$$\bar{A}^{\pi_{\theta_k}}(s,a) = r(s,a) - \rho(\pi_{\theta_k}) + \mathop{\mathbb{E}}_{s'\sim P(\cdot|s,a)} \left[\bar{V}^{\pi_{\theta_k}}(s')\right] - \bar{V}^{\pi_{\theta_k}}(s) \tag{44}$$

This expression involves the average-reward bias $\bar{V}^{\pi_{\theta_k}}(s)$, which we can approximate using a critic network $\bar{V}_{\phi_k}(s)$, giving line 7 in Algorithm 2. It remains to specify what the target should be for updating the critic parameter ϕ. For this, we can similarly make use of the Bellman equation for the average-reward bias in Equation (15) which gives line 6 in Algorithm 2. Finally, like discounted TRPO, after applying the update term (42), we use backtracking linesearch to find an update term which has a positive advantage value and also maintains KL constraint satisfaction. We also apply the conjugate gradient method to estimate H^{-1}.

F. Reinforcement Learning with Average Cost Constraints

F.1. The Constrained RL Problem

In addition to learning to improve its long-term performance, many real-world 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 $\{\pi \in \Pi : \rho_c(\pi) \leqslant b\}$ and the discounted cost constraint set $\{\pi \in \Pi : \rho_{c,\gamma}(\pi) \leqslant b\}$. Here b is some given constraint bound, the cost constraint functions are given by

$$\rho_c(\pi) := \lim_{N\to\infty} \frac{1}{N} \mathop{\mathbb{E}}_{\tau\sim\pi} \left[\sum_{t=0}^{N-1} c(s_t, a_t)\right] \tag{45}$$

$$\rho_{c,\gamma}(\pi) := \mathop{\mathbb{E}}_{\tau\sim\pi} \left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t)\right] \tag{46}$$

for some bounded cost function $c : \mathcal{S} \times \mathcal{A} \to [c_{\min}, c_{\max}]$.

F.2. Constrained RL via Local Policy Update

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:

$$\begin{aligned} \underset{\pi_\theta \in \Pi_\theta}{\text{maximize}} \quad & \underset{\substack{s\sim d_{\pi_{\theta_k},\gamma} \\ a\sim\pi_\theta}}{\mathbb{E}} [A_\gamma^{\pi_{\theta_k}}(s,a)] \\ \text{subject to} \quad & \tilde{\rho}_{c,\gamma}(\pi_\theta) \leqslant b, \\ & \bar{D}_{\text{KL}}(\pi_\theta \parallel \pi_{\theta_k}) \leqslant \delta. \end{aligned} \tag{47}$$

Here,

$$\tilde{\rho}_{c,\gamma}(\pi_\theta) := \rho_{c,\gamma}(\pi_{\theta_k}) + \frac{1}{1-\gamma} \underset{s\sim d_{\pi_{\theta_k},\gamma}, a\sim\pi_\theta}{\mathbb{E}} \left[A_{c,\gamma}^{\pi_{\theta_k}}(s,a) \right] \tag{48}$$

is a surrogate cost function used to approximate the cost constraint and $A_{c,\gamma}^{\pi_{\theta_k}}(s,a)$ is the discounted cost advantage function where we replace the reward with the cost[2]. Note that (48) can be evaluated using samples from π_{θ_k}. By Corollary 2 of Achiam et al. (2017) and (11):

$$|\rho_{c,\gamma}(\pi_\theta) - \tilde{\rho}_{c,\gamma}(\pi_\theta)| \leqslant \frac{\gamma \epsilon_{c,\gamma}}{(1-\gamma)^2} \sqrt{2 \bar{D}_{\text{KL}}(\pi_\theta \parallel \pi_{\theta_k})} \tag{49}$$

where $\epsilon_{c,\gamma} = \max_s \left| \mathbb{E}_{a\sim\pi'}[\bar{A}_{c,\gamma}^{\pi}(s,a)] \right|$. 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 w.r.t. the KL divergence. Using (49) and the trust region constraint, the worst-case constraint violation for when $\pi_{\theta_{k+1}}$ is the solution to (47) can be upper bounded (Proposition 2 of Achiam et al. (2017).)

This framework is problematic when the cost constraint is undiscounted. Define the average surrogate cost as

$$\tilde{\rho}_c(\pi_\theta) := \rho_c(\pi_{\theta_k}) + \underset{\substack{s\sim d_{\pi_{\theta_k}} \\ a\sim\pi_{\theta_k}}}{\mathbb{E}} [A_c^{\pi_\theta}(s,a)] \tag{50}$$

where $A_c^{\pi_\theta}(s,a)$ is the average cost advantage function. 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[3] and (11):

$$|\rho_c(\pi_\theta) - \tilde{\rho}_c(\pi_\theta)| \leqslant \xi_c^{\pi_\theta} \sqrt{2 \bar{D}_{\text{KL}}(\pi_\theta \parallel \pi_{\theta_k})} \tag{51}$$

where $\xi_c^{\pi_\theta} = (\kappa^\star - 1) \max_s \mathbb{E}_{a\sim\pi_\theta} |A_c^{\pi_{\theta_k}}(s,a)|$. We then have the following result:

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

$$\rho_C(\pi_\theta) \leqslant b + \xi_c^{\pi_\theta} \sqrt{2\delta} \tag{52}$$

The upper-bound in Proposition F.1 provides a worst-case constraint violation guarantee when π_θ is the solution to the average-cost variant of (47). 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), (47) 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 (47) (Achiam et al., 2017; Zhang et al., 2020). Proposition F.1 shows that these algorithms can be generalized to average cost constraints with only minor modifications. In the next section, we will show how the CPO algorithm (Achiam et al., 2017) can be modified for average cost constraints.

[2] We can define the discounted value/action-value cost functions and the average cost bias/action-bias functions in a similar manner

[3] It is straightforward to show that the theorem still holds when we replace the reward with the cost.

F.3. Average Cost CPO (ACPO)

Consider the average cost variant of (47):

$$\begin{aligned} \underset{\pi_\theta \in \Pi_\theta}{\text{maximize}} \quad & \underset{\substack{s\sim d_{\pi_{\theta_k}} \\ a\sim\pi_\theta}}{\mathbb{E}} \left[\bar{A}^{\pi_{\theta_k}}(s,a)\right] \\ \text{subject to} \quad & \tilde{\rho}_c(\pi_\theta) \leqslant b, \\ & \bar{D}_{\text{KL}}(\pi_\theta \parallel \pi_{\theta_k}) \leqslant \delta. \end{aligned} \tag{53}$$

Similar to TRPO/ATRPO, we can apply first and second order Taylor approximations to (53) which then gives us

$$\begin{aligned} \underset{\theta}{\text{maximize}} \quad & g^T(\theta - \theta_k) \\ \text{subject to} \quad & \tilde{c} + \tilde{g}^T(\theta - \theta_k) \leqslant 0 \\ & \frac{1}{2}(\theta - \theta_k)^T H(\theta - \theta_k) \leqslant \delta \end{aligned} \tag{54}$$

where g, H were defined in the previous section, $\tilde{c} = \rho(\pi_{\theta_k}) - b$, and

$$\tilde{g} := \underset{\substack{s\sim d_{\pi_{\theta_k}} \\ a\sim\pi_{\theta_k}}}{\mathbb{E}} \left[\nabla_\theta \log \pi_\theta(a|s)|_{\theta=\theta_k} \bar{A}_c^{\pi_{\theta_k}}(s,a)\right] \tag{55}$$

is the gradient of the constraint. Similar to the case of ATRPO, g, $\tilde{g}$, H, and $\tilde{c}$ can all be approximated using samples collected from π_{θ_k}. The term $\bar{A}_c^{\pi_{\theta_k}}(s,a)$ also involves the cost bias-function (see Equation 44) which can be approximated via a separate cost critic network. The optimization problem (54) 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

$$\theta = \theta_k + \frac{1}{\lambda} H^{-1}(g - \nu\tilde{g}) \tag{56}$$

where λ and ν are Lagrange multipliers satisfying (Achiam et al., 2017)

$$\max_{\lambda,\nu\geqslant 0} -\frac{1}{2\lambda}\left(g^T H^{-1} g + 2\nu g^T H^{-1}\tilde{g} + \nu^2 \tilde{g} H^{-1}\tilde{g}^T\right) + \nu\tilde{c} - \frac{1}{2}\lambda\delta \tag{57}$$

The dual problem (57) 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.4. Experiment Results

For constrained RL algorithms, being able to accurately evaluate the cost constraint for a particular policy is key to learning constraint-satisfying policies. In this section, we consider the MuJoCo agents from Section 6. However for safety reasons, we wish the agent to maintain its average speed over a trajectory below a certain threshold which is set at 2.0 for all environments.

Here we use the same evaluation protocol as was introduced in Section 6.1 but we calculated the average cost (total cost / trajectory length) as well as the total return for each evaluation trajectory. We used a maximum trajectory length of 1000 for these experiments. We plotted the results of our experiments in Figure 3.

From Figure 3 we see that ACPO is able to learn high-performing policies while enforcing the average cost constraint.

G. Generalized Advantage Estimator (GAE) 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 V_ϕ^π 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 V_ϕ^π are the *Monte Carlo* target denoted by

$$V_t^{\text{target}} = \sum_{t'=t}^{N} \gamma^{t'-t} r_t, \tag{58}$$

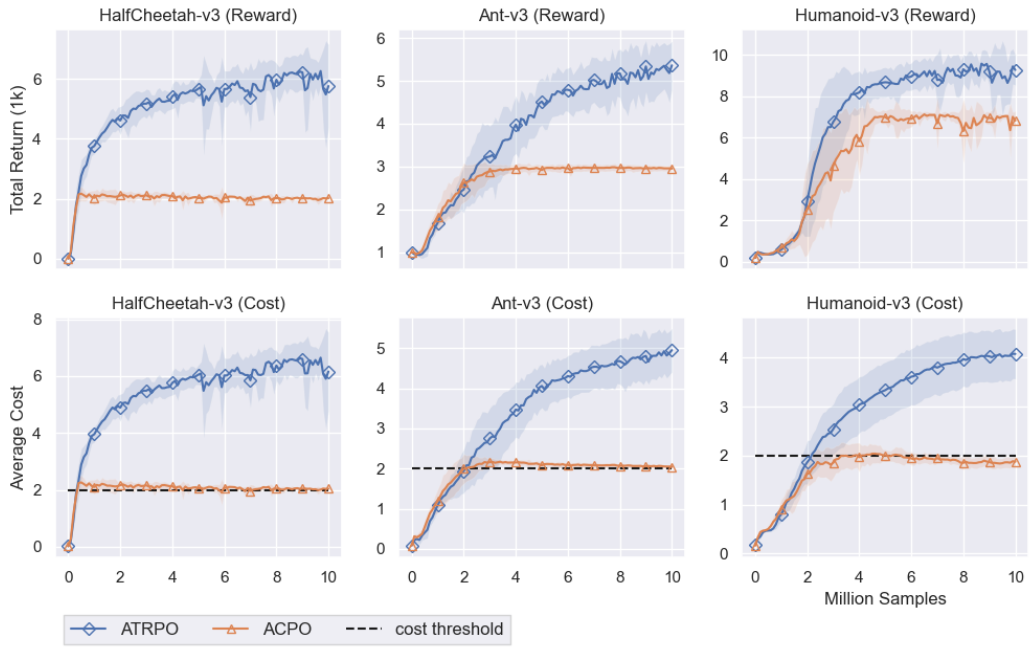

Figure 3. Performance for ACPO. Unconstrained ATRPO is plotted for comparison. The x-axis is the number of agent-environment interactions and the y-axis is the total return averaged over 10 seeds. The solid line represents the agents' average total return (top row) and average cost (bottom row) on the evaluation trajectories. The shaded region represent one standard deviation.

or the *bootstrapped* target

$$V_t^{\text{target}} = r_t + \gamma \bar{V}_\phi^\pi(s_{t+1}). \tag{59}$$

Using the dataset $\{s_t, V_t^{\text{target}}\}$, we can fit V_ϕ^π with supervised regression by minimizing the MSE between $V_\phi^\pi(s_t)$ and V_t^{target}. With the fitted value function, we can estimate the advantage function either with the Monte Carlo estimator

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

or the bootstrap estimator

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

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

$$\hat{A}_{\text{GAE}}(s_t, a_t) = \sum_{t'=t}^{N} (\gamma\lambda)^{t'-t} \delta_{t'} \tag{60}$$

where

$$\delta_{t'} = r_{t'} + \gamma \bar{V}_\phi^\pi(s_{t'+1}) - \bar{V}_\phi^\pi(s_{t'}) \tag{61}$$

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:

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

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

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 is that in the discounted case we use V_ϕ^π to approximate the *discounted value function* whereas in the average reward case $\bar{V}_\phi^\pi$ is used to approximate the *average reward bias function*.

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

$$\bar{V}_t^{\text{target}} = \sum_{t'=t}^{N} (r_t - \hat{\rho}_\pi) \tag{62}$$

and the bootstrapped target is

$$\bar{V}_t^{\text{target}} = r_t - \hat{\rho}_\pi + \bar{V}_\phi^\pi(s_{t+1}). \tag{63}$$

Note that our targets (62-63) are distinctly different from the traditional discounted targets (58-59).

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

$$\hat{A}_{\text{MC}}^\pi(s_t, a_t) = \sum_{t'=t}^{N} (r_t - \hat{\rho}_\pi) - \bar{V}_\phi^\pi(s_t)$$
$$\hat{A}_{\text{BS}}^\pi(s_t, a_t) = r_{i,t} - \hat{\rho}_\pi + \bar{V}_\phi^\pi(s_{t+1}) - \bar{V}_\phi^\pi(s_t)$$

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

$$\hat{A}_{\text{GAE}}(s_t, a_t) = \sum_{t'=t}^{N} \lambda^{t'-t}\delta_{t'} \tag{64}$$

where

$$\delta_{t'} = r_{t'} - \hat{\rho}_\pi + \bar{V}_\phi^\pi(s_{t'+1}) - \bar{V}_\phi^\pi(s_{t'}). \tag{65}$$

and set the target for the value function to

$$\bar{V}_t^{\text{target}} = r_t - \hat{\rho}_\pi + \bar{V}_\phi^\pi(s_{t+1}) + \sum_{t'=t+1}^{N} \lambda^{t'-t}\delta_{t'} \tag{66}$$

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

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

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 bias 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.

H. Experimental Setup

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 and https://github.com/openai/safety-starter-agents. 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.

With the exception of results in Appendix I.2, the reset cost is set to 100 on all three environments. In the original implementation of the MuJoCo environments in OpenAI gym, the maximum episode length is set to 1000[4], we removed this restriction in our experiments in order to study long-run performance.

We used a two-layer feedforward neural network with a $\tanh$ activation for both our policy and critic 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 G). The advantage values are normalized by its batch mean and batch standard deviation before being used for policy updates. Learning rates are linearly annealed to 0 over the course of training. Note that these settings are common in most open-source implementations of TRPO and other on-policy algorithms. For training and evaluation, we used different random seeds (i.e. the random seeds we used to generate the evaluation trajectories are different from those used during training.) Table 2 summarizes the hyperparameters used in our experiments.

Table 2. Hyperparameter Setup

Hyperparameter	TRPO/ATRPO	CPO/ACPO
No. of hidden layers	2	2
No. of hidden nodes	64	64
Activation	$\tanh$	$\tanh$
Initial log std	-0.5	-1
Batch size	5000	5000
GAE parameter (reward)	0.95	0.95
GAE parameter (cost)	N/A	0.95
Learning rate for policy	3×10^{-4}	3×10^{-4}
Learning rate for reward critic net	3×10^{-4}	3×10^{-4}
Learning rate for cost critic net	N/A	3×10^{-4}
$L2$-regularization coeff. for critc net	3×10^{-3}	3×10^{-3}
Damping coeff.	0.01	0.01
Backtracking coeff.	0.8	0.8
Max backtracking iterations	10	10
Max conjugate gradient iterations	10	10
Trust region bound δ	0.01	0.01

I. Additional Experiments

I.1. Comparing with TRPO Trained Without Resets

Figure 4 repeats the experiments presented in Figure 1 except discounted TRPO is trained in the standard MuJoCo setting without any resets (i.e. during training, when the agent falls, the trajectory terminates.) The maximum length of a TRPO training episode is 1000. This is identical to how TRPO is trained in the literature for the MuJoCo environments. We apply

[4]See https://github.com/openai/gym/blob/master/gym/envs/__init__.py

the same evaluation protocol introduced in Section 6.1. We note that when TRPO is trained in the standard MuJoCo setting, ATRPO still outperforms discounted TRPO by a significant margin.

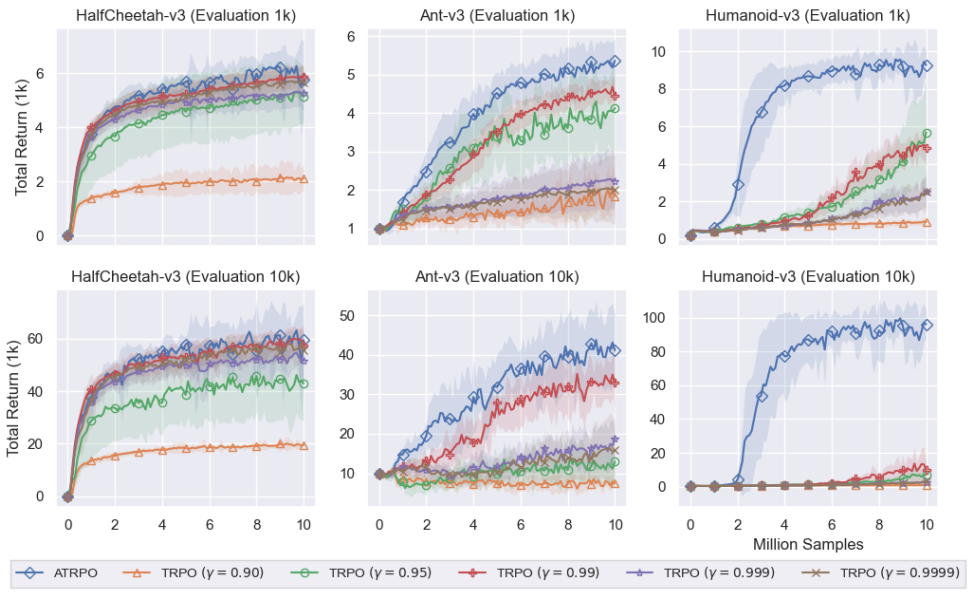

Figure 4. Comparing performance of ATRPO and TRPO with different discount factors. TRPO is trained without the reset scheme. The x-axis is the number of agent-environment interactions and the y-axis is the total return averaged over 10 seeds. The solid line represents the agents' performance on evaluation trajectories of maximum length 1,000 (top row) and 10,000 (bottom row). The shaded region represent one standard deviation.

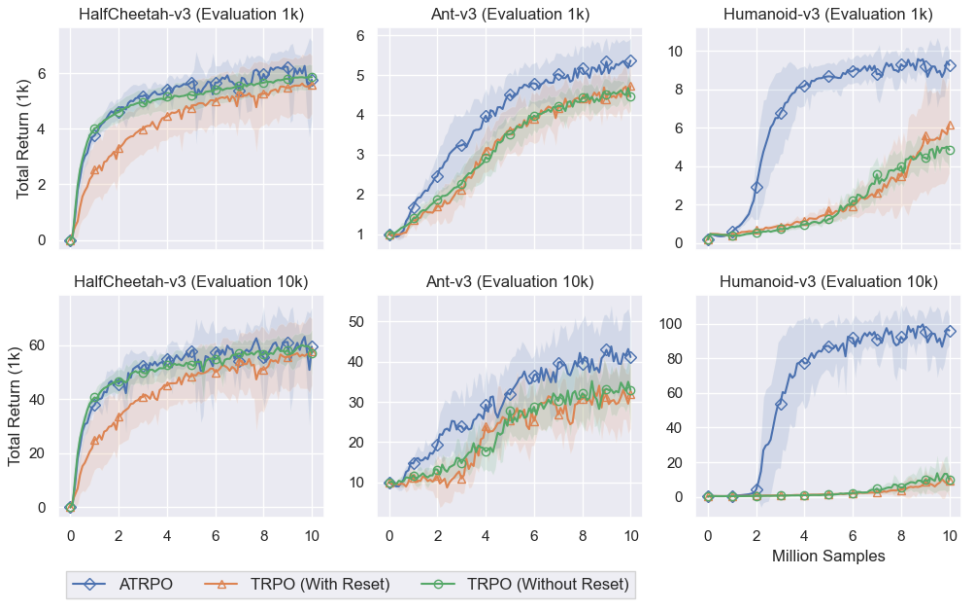

Figure 5. Comparing performance of ATRPO and TRPO trained with and without the reset costs. The curves for TRPO are for the best discount factor for each environment. The x-axis is the number of agent-environment interactions and the y-axis is the total return averaged over 10 seeds. The solid line represents the agents' performance on evaluation trajectories of maximum length 1,000 (top row) and 10,000 (bottom row). The shaded region represent one standard deviation.

In Figure 5 we plotted the performance of the best discount factor for each environment for TRPO trained with and without the reset scheme (i.e. the best performing TRPO curves from Figure 1 and Figure 4.) ATRPO is also plotted for comparison.

We note here that the performance of TRPO trained with and without the reset scheme are quite similar, this further supports the notion that introducing the reset scheme does not alter the goal of the tasks.

I.2. Sensitivity Analysis on Reset Cost

For the experiments presented in Figure 2, we introduced a reset cost in order to simulate an infinite horizon setting. Here we analyze the sensitivity of the results with respect to this reset cost.

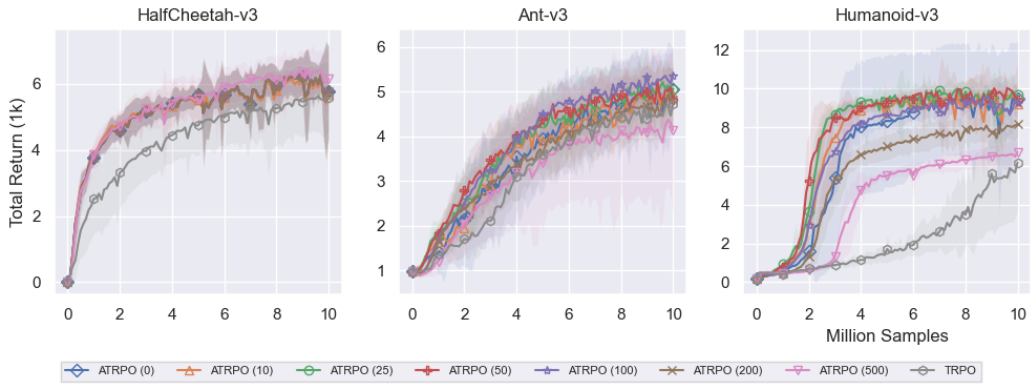

Figure 6. Comparing ATRPO trained with different reset costs to discounted TRPO with the best discount factor for each environment. The x-axis is the number of agent-environment interactions and the y-axis is the total return averaged over 10 seeds. The solid line represents the agents' performance on evaluation trajectories of maximum length 1,000. The shaded region represent one standard deviation.

Figure 6 shows that ATRPO is largely insensitive to the choice of reset cost. Though we note that for Humanoid, extremely large reset costs (200 and 500) does negatively impact performance but the result is still above that of TRPO.