On the Sample Complexity of Actor-Critic for Reinforcement Learning

Harshat Kumar
Dept. of Electrical and Systems Eng.
University of Pennsylvania
Philadelphia, PA 19104
harshat@seas.upenn.edu

Alec Koppel
Computational and Information Sciences Directorate
U.S. Army Research Laboratory
Aldephi, MD 20783
alec.e.koppel.civ@mail.mil

Alejandro Ribeiro
Dept. of Electrical and Systems Eng.
University of Pennsylvania
Philadelphia, PA 19104
aribeiro@seas.upenn.edu

Abstract

Combining dynamic programming and policy search techniques to solve reinforcement learning problems, actor-critic algorithms are known to work well in practice. Despite its widespread use, only the asymptotic behavior of actor-critic is known by connecting its behavior to dynamical systems. This work attempts to close this gap in existing literature by characterizing the rate of convergence in expectation for actor-critic in continuous state action space with linear function approximation using policy gradient for policy search and any critic method. Our rates are corroborated by implementing actor critic on a navigation problem with Temporal Difference, Gradient Temporal Difference, and Accelerated Gradient Temporal Difference critic updates.

1 Introduction

Reinforcement learning (RL) is a form of adaptive control where the system model is unknown and one seeks to estimate parameters of a controller through repeated interaction with the environment [3, 26]. This framework gained attention recently for its ability to express problems that exhibit complicated dependencies between action selection and environmental response, i.e., when the cost function or system dynamics are difficult to express, as in supply chain management [14], power systems [15], robotic manipulation [16], and games of various kinds [29, 24, 9]. Although the expressive capability of RL continues to motivate new and diverse applications, its computational challenges remain doggedly persistent.

Two dominant approaches to RL have emerged since its original conception from Bellman [2]. The first, dynamic programming [33], writes the value as the expected one-step reward plus all subsequent rewards (Bellman equations), and then proceeds by stochastic fixed point iterations [30]. Combining dynamic programming approaches with nonlinear function parameterizations, as noted by [31], may cause instability. On the other hand, the alternative approach, policy search [27], hypothesizes actions are chosen according to a parameterized distribution, and then repeatedly revises those parameters according via stochastic search. Policy search has gained popularity due to its ability to scale to large (or continuous) spaces and exhibit global convergence. Also worth mentioning is Monte Carlo search ("guess and check") [1, 13], which is essential to reducing large spaces to only viable hypotheses.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

In this work, we focus on methods that operate in the intersection of dynamic programming and policy search called *actor-critic* [18, 17]. Actor-critic is an online form of policy iteration [3] that inherits the ability of policy search to scale to large (or continuous) spaces, while reducing its number of queries to the environment. In particular, policy gradient method repeatedly revises policy parameter estimates through gradient steps. Via the Policy Gradient Theorem [27], the policy gradient is the product of two factors: the score function and the Q function. One may employ Monte Carlo rollouts to acquire the Q-estimates, which under careful choice of the rollout horizon, can be shown to be unbiased [22, 34]. Doing so, however, requires an inordinate amount of querying to the environment.

Actor-critic replaces Monte-Carlo rollouts for the Q-value by stochastic approximates of solutions to Bellman equations, i.e., temporal difference (TD) [25] or gradient temporal difference (GTD) [28] steps. Intuitively, this weaving together of the merits of dynamic programming and policy search yields comparable scalability properties to policy search while reducing its sample complexity. However, the iteration (and sample) complexity of actor-critic is noticeably absent from the literature, which is striking due to its foundational role in modern reinforcement learning systems [20, 24], and the fact that efforts to improve upon it also only establish asymptotics [6].

Therefore, in this work, we characterize the complexity of actor-critic algorithms. Our results are framed by viewing policy search as a form of stochastic gradient method for maximizing a non-convex function, where the ascent directions are biased. Moreover, the magnitude of this bias is determined the number of critic steps. This perspective treats actor-critic a form of two time-scale algorithm [7], whose asymptotic stability is well-known via dynamical systems tools [19, 8]. To wield these approaches to establish finite-time performance, however, concentration probabilities and geometric ergodicity assumptions of the Markov dynamics are required – see [8]. To obviate these complications and exploit recent unbiased sampling procedures [22, 34], we focus on the case where independent trajectory samples are acquirable through querying the environment.

Our main result establishes that actor-critic, independent of any critic method, exhibits convergence to stationary points of the value function that are comparable to stochastic gradient ascent in the non-convex regime. We note that a key distinguishing feature from standard non-convex stochastic programming is that the rates are inherently tied to the bias of the search direction which is determined by the choice of critic scheme. The rate we derive (See Theorem 1) depends on the convergence rate of the critic method. Critic methods with slow convergence become the bottleneck for actor-critic, while methods with fast convergence guarantees shift bottleneck to the policy gradient updates. Empirically, we show that although fast critic convergence will result in faster convergence for actor-critic, the stationary point it reaches is worse than methods which have slower convergence guarantees. Experiments on a simple navigation problem with an obstacle demonstrate the empirical validity of these results, which overall illuminate the interplay between generalization, optimization, and function parameterization in reinforcement learning.

2 Reinforcement Learning

In reinforcement learning (RL), an agent moves through a state space $\mathcal{S}$ and takes actions belonging to set $\mathcal{A}$, where the state/action spaces are assumed to be continuous compact subsets of Euclidean space: $\mathcal{S} \subset \mathbb{R}^q$ and $\mathcal{A} \subset \mathbb{R}^p$. After each action, the agent transitions to its next state that depending on its current state and action. Moreover, a reward is revealed by the environment. In this situation, the agent would like to accumulate as much reward as possible in the long term, which is referred to as value. Mathematically this problem definition may be encapsulated as a Markov decision process (MDP), which is a a tuple $(\mathcal{S}, \mathcal{A}, \mathbb{P}, R, \gamma)$ with Markov transition density $\mathbb{P}(s' \mid s, a) : \mathcal{S} \times \mathcal{A} \to \mathbb{P}(\mathcal{S})$ that determines the probability of moving to state s'. Here, $\gamma \in (0, 1)$ is the discount factor that parameterizes the value of a given sequence of actions, which we will define shortly.

At each time t, the agent executes an action $a_t \in \mathcal{A}$ given the current state $s_t \in \mathcal{S}$, following a possibly stochastic policy $\pi : \mathcal{S} \to \mathbb{P}(\mathcal{A})$, i.e., $a_t \sim \pi(\cdot \mid s_t)$. Then, given the state-action pair (s_t, a_t), the agent observes a (deterministic) reward $r_t = R(s_t, a_t)$ and transitions to a new state $s'_t \sim \mathbb{P}(\cdot \mid s_t, a_t)$ according to a transition density that is Markov. For any policy π mapping states to actions, define the value function $V_\pi : \mathcal{S} \to \mathbb{R}$ as

$$V_\pi(s) = \mathbb{E}_{a_t \sim \pi(\cdot|s_t), s_{t+1} \sim \mathbb{P}(\cdot|s_t, a_t)} \left(\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s \right), \tag{1}$$

which is the long term average reward accumulation discounted by γ. We can further define the value $V_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ conditioned on a given initial action as the action-value, or Q-function as $Q_\pi(s,a) = \mathbb{E}\big(\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s, a_0 = a\big)$. Given initial state s_0, the agent seeks to find policy π that maximizes the long-term return $V_\pi(s_0)$, i.e., to solve the following optimization problem

$$\max_{\pi \in \Pi} \; J(\pi) := V_\pi(s_0). \tag{2}$$

In this work, we investigate actor-critic methods to solve (2), which is a hybrid RL method that fuses key properties of policy search and approximate dynamic programming. We first derive policy gradient method, and explain how actor-critic augments policy search. To mitigate the issue of searching over an arbitrarily complicated function class Π, we parameterize the policy π by a vector $\theta \in \mathbb{R}^d$, i.e., $\pi = \pi_\theta$, as in [17, 6, 10]. Under this specification, the search may be reduced to vector-valued optimization $\max_{\theta \in \mathbb{R}^d} J(\theta) := V_{\pi_\theta}(s_0)$. Subsequently, we denote $J(\pi_\theta)$ by $J(\theta)$ for notational convenience. We first make the following standard assumption on the regularity of the MDP problem and the parameterized policy π_θ, which are the same conditions as [35].

Assumption 1. *Suppose the reward R and the parameterized policy π_θ satisfy the following:*

(i) The reward R is bounded absolutely and uniformly: $|R(s,a)| \in [0, U_R]$ for $(s,a) \in \mathcal{S} \times \mathcal{A}$.

(ii) The policy π_θ is differentiable with respect to θ, and the score function $\nabla \log \pi_\theta(a \mid s)$ is L_Θ-Lipschitz and has bounded norm, i.e., for any $(s,a) \in \mathcal{S} \times \mathcal{A}$,

$$\|\nabla \log \pi_{\theta^1}(a \mid s) - \nabla \log \pi_{\theta^2}(a \mid s)\| \le L_\Theta \cdot \|\theta^1 - \theta^2\|, \;\; \textit{for any} \;\; \theta^1, \theta^2, \tag{3}$$

$$\|\nabla \log \pi_\theta(a \mid s)\| \le B_\Theta, \;\; \textit{for any} \;\; \theta. \tag{4}$$

Note that the boundedness of the reward function in Assumption1(i) is standard in policy search algorithms [5, 6, 10, 36]. Observe that with R, we have the Q-function is absolutely upper bounded:

$$|Q_{\pi_\theta}(s,a)| \le \sum_{t=0}^{\infty} \gamma^t \cdot U_R = U_R/(1-\gamma), \;\; \text{for any} \;\; (s,a) \in \mathcal{S} \times \mathcal{A}. \tag{5}$$

The same holds true for $V_{\pi_\theta}(s)$ for any π_θ and $s \in \mathcal{S}$ and thus for the objective $J(\theta) := V_{\pi_\theta}(s_0)$, i.e.,

$$|V_{\pi_\theta}(s)| \le U_R/(1-\gamma), \;\; \text{for any } s \in \mathcal{S}, \qquad |J(\theta)| \le U_R/(1-\gamma). \tag{6}$$

We note that the conditions (3) and (4) have appeared in recent analyses of policy search [10, 23, 21], and are satisfied by canonical policy parameterizations such as Boltzmann policy [18] and Gaussian policy [12]. For example, for Gaussian policy[1] in continuous spaces, $\pi_\theta(\cdot \mid s) = \mathcal{N}(\phi(s)^\top \theta, \sigma^2)$, where $\mathcal{N}(\mu, \sigma^2)$ denotes the Gaussian distribution with mean μ and variance σ^2. Then the score function has the form $[a - \phi(s)^\top \theta]\phi(s)/\sigma^2$, which satisfies (3) and (4) if the feature vectors $\phi(s)$ have bounded norm, the parameter θ lies some bounded set, and the action $a \in \mathcal{A}$ is bounded.

Assumption 2. *For any state action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$, the feature map $\varphi(s,a)$ has norm bounded by a constant $C_2 \in \mathbb{R}_+$.*

The score function is bounded by B_Θ (c.f. Assumption 1) and reward is bounded, so for some constant $C_3 \in \mathbb{R}_+$

$$\|J(\theta_k)\| \le C_3 \text{for all } (s,a) \in \mathcal{S} \times \mathcal{A} \tag{7}$$

We further assume the estimate of the Q function conditioned on the filtration has finite variance σ^2.

Assumption 3. *There exits a $\sigma^2 > 0$ such that for all state action pairs $(s_t, a_t) \in \mathcal{S} \times \mathcal{A}$,*

$$\mathbb{E}(\|\xi_t^\top \varphi(s_t, a_t) \nabla \log \pi(s_t, a_t | \theta_k)\| | \mathcal{F}_k) \le \sigma^2. \tag{8}$$

Our goal is to design actor-critic algorithms to attain stationary points of the value function $J(\theta)$. Moreover, we characterize the sample complexity of actor-critic, a noticeable gap in the literature for an algorithmic tool decades old [18] at the heart of the recent artificial intelligence innovations [24].

[1] In practice, the action space $\mathcal{A}$ is bounded, which requires a truncated Gaussian policy over $\mathcal{A}$, as in [21].

3 From Policy Gradient to Actor-Critic

In this section, we derive actor-critic method [18] from an optimization perspective: we view actor-critic as a way of doing stochastic gradient ascent with biased ascent directions, and the magnitude of this bias is determined by the number of critic evaluations done in the inner loop of the algorithm. The building block of actor-critic is called policy gradient method, a type of direct policy search, based on stochastic gradient ascent. Begin by noting that the gradient of the objective $J(\theta)$ with respect to policy parameters θ, owing to the Policy Gradient Theorem [27], has the following form:

$$\nabla J(\theta) = \frac{1}{1-\gamma} \cdot \mathbb{E}_{(s,a)\sim\rho_\theta(\cdot,\cdot)}\big[\nabla \log \pi_\theta(a \mid s) \cdot Q_{\pi_\theta}(s,a)\big]. \tag{9}$$

The derivative of the logarithm of the policy $\nabla \log[\pi_\theta(\cdot \mid s)]$ is referred to as the *score function* corresponding to the policy $\pi_\theta(\cdot \mid s)$ for any $s \in \mathcal{S}$. Unbiased samples of the gradient $\nabla J(\theta)$ are required to perform the stochastic gradient ascent, which hopefully converges to a stationary solution of the nonconvex maximization. One may obtain an estimate of the gradient $\nabla J(\theta)$ by evaluating the score function and Q function at the end of a rollout whose length is drawn from a geometric distribution with parameter $1-\gamma$ [35][Theorem 4.3]. If the Q estimate is unbiased, then the stochastic estimate of the gradient $\nabla J(\theta)$ is unbiased as well. Therefore define the stochastic estimate by

$$\hat{\nabla} J(\theta) = \frac{1}{1-\gamma} \hat{Q}_{\pi_\theta}(s_T, a_T) \nabla \log \pi_\theta(a_T|s_T). \tag{10}$$

We consider the case where the Q function admits a linear parametrization of the form $\hat{Q}_{\pi_\theta}(s,a) = \xi^\top \varphi(s,a)$, which in the literature on policy search is referred to as the *critic* [18], as it "criticizes" the performance of actions chosen according to policy π. Here $\xi \in \mathbb{R}^p$ and $\varphi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^p$ is a (possibly nonlinear) feature map such as a network of radial basis functions [] or auto-encoder []. Moreover, we estimate the parameter ξ that defines the Q function from a policy evaluation (critic only) method after some $T_C(k)$ iterations, where k denotes the number of policy gradient updates. Thus, we may write the stochastic gradient estimate as

$$\hat{\nabla} J(\theta) = \frac{1}{1-\gamma} \xi_k^\top \varphi(s_T, a_T) \nabla \log \pi_\theta(a_T|s_T). \tag{11}$$

If the estimate of the Q function is unbiased, i.e., $\mathbb{E}[\xi_k^\top \varphi(s_T, a_T) \,|\, \theta, s, a] = Q(s,a)$, then $\mathbb{E}[\hat{\nabla} J(\theta) \,|\, \theta] = \nabla J(\theta)$ (c.f. [35][Theorem 4.3]). Typically, critic only methods do not give unbiased estimates of the Q function; however, in expectation the rate at which their bias decays is proportional to the number of Q estimation steps. In particular, denote ξ_* as the parameter for which the Q estimate is unbiased:

$$\mathbb{E}[\xi_*^\top \varphi(s,a)] = \mathbb{E}[\hat{Q}_{\pi_\theta}(s,a)] = Q(s,a). \tag{12}$$

Hence, by adding and subtracting the true estimate of the parametrized Q function to (11), we arrive at the fact the policy search direction admits the following decomposition:

$$\hat{\nabla} J(\theta) = \frac{1}{1-\gamma} (\xi_k - \xi_*)^\top \varphi(s_T, a_T) \nabla \log \pi_\theta(a_T|s_T) + \frac{1}{1-\gamma} (\xi_*)^\top \varphi(s_T, a_T) \nabla \log \pi_\theta(a_T|s_T). \tag{13}$$

The second term in is the unbiased estimate of the gradient $\nabla J(x)$, whereas the first defines the difference of the critic parameter at iteration k with the true estimate ξ_*. For linear parameterizations of the Q function, policy evaluation methods establish convergence in mean of the bias $\mathbb{E}[\|\xi_k - \xi_*\|] \leq g(k)$, where $g(k)$ is some decreasing function. We address cases where the critic bias decays at rate k^{-b} for $b > 0$, due to the fact that several state of the art works on policy evaluation may be mapped to the form (14) for this specification [32, 11, 4, 37].

Assumption 4. *The critic parameter has error mean bounded by $O(k^{-b})$ for some $b \in (0,1]$, i.e., there exists constant $L_1 > 0$ such that*

$$\mathbb{E}[\|\xi_k - \xi_*\|] \leq L_1 k^{-b}. \tag{14}$$

Several existing bounds on the policy evaluation error take the form k^{-b} for some $b \in (0,1]$ [32, 11, 4, 37]. Alternate rates have been established as $O(\log k/k)$; however, they concede that

Algorithm 1 Generic Actor-Critic

Require:

$s_0 \in \mathbb{R}^n$, θ_0, ξ_0, stepsize $\{\eta_k\}$, Policy evaluation method **Critic**: $\mathbb{N} \to \mathbb{R}^p$, $\gamma \in (0,1)$

1: **for** $k = 1, \dots$ **do**
2: $\quad \xi_{T_C(k)} \leftarrow$ **Critic**$(T_C(k))$
3: $\quad$ Draw $T_k \sim$ **Geom**$(1-\gamma)$
4: $\quad (s_{T_k}, a_{T_k}) \leftarrow$ rollout of π_{θ_k} with length T_k
5: $\quad \theta_{k+1} \leftarrow \theta_k + \frac{1}{1-\gamma}\eta_k \xi_{T_C(k)}^\top \varphi(s_{T_k}, a_{T_k}) \nabla \log \pi_{\theta_k}(s_{T_k}, a_{T_k}|\theta_k)$

$O(1/k)$ rates may be possible. Thus, we subsume sample complexity characterizations of policy evaluation as (14).

Thus, (13) is nearly a valid ascent direction: it is approximately an unbiased estimate of the gradient $\nabla J(\theta)$ since the first term becomes negligible as the number of critic estimation steps increases. Based upon this observation, we propose the following variant of actor-critic method [18]: run a critic estimator (policy evaluator) for $T_C(k)$ steps, whose output is critic parameters $\xi_{T_C(k)}$. We denote the critic estimator by **Critic:**$\mathbb{N} \to \mathbb{R}^p$ which returns the parameter $\xi_{T_C(k)} \in \mathbb{R}^p$ after $T_C(k) \in \mathbb{N}$ iterations. Then, simulate a trajectory of length T_k, where T_k is geometrically distributed with parameter $1-\gamma$, and update the actor (policy) parameters θ as:

$$\theta_{k+1} = \theta_k + \eta_k \frac{1}{1-\gamma} \eta_k \xi_{T_C(k)}^\top \varphi(s_{T_k}, a_{T_k}) \nabla \log \pi_{\theta_k}(s_{T_k}, a_{T_k}|\theta_k) \tag{15}$$

We summarize the aforementioned procedure, which is agnostic to particular choice of critic estimator, as Algorithm 1. Subsequently, we shift focus to characterizing the mean convergence of actor-critic method given any policy evaluation method satisfying (14). By Assumption 4 (which is not really an assumption but rather a fundamental property of most common policy evaluation schemes), the error goes to zero in expectation as the number of critic steps increases. Thus, we leverage this property to derive the sample complexity of actor-critic (Algorithm 1). To do so, we define the rate in terms of the complexity measure K_ϵ, the smallest number of actor updates k required to attain a value function gradient smaller ϵ, i,.e.,

$$K_\epsilon = \min\{k : \inf_{0 \le m \le k} \|\nabla J(\theta_m)\|^2 < \epsilon\}. \tag{16}$$

Theorem 1. Suppose the step-size satisfies $\eta_k = k^{-a}$ for $a > 0$ and the critic update satisfies Assumption 4. When the critic bias converges to null as $\mathcal{O}(k^{-1})$, i.e., $b = 1$ in (14), then $T_C(k) = k+1$ critic updates occur per actor update. Alternatively, if the critic bias converges to null more slowly $b \in (0,1)$ then $T_C(k) = k$ critic updates per actor update are chosen. Then the actor sequence defined by Algorithm 1 satisfies

$$K_\epsilon \le \mathcal{O}\left(\epsilon^{-1/\ell}\right) \text{ , where } \ell = \min\{a, 1-a, b\} \tag{17}$$

Minimizing over a yields actor step-size $\eta_k = k^{-1/2}$. Moreover, depending on the rate b of attenuation of the critic bias [cf. (14)], the resulting sample complexity is:

$$K_\epsilon \le \begin{cases} \mathcal{O}\left(\epsilon^{-1/b}\right) & \text{if } b \in (0, 1/2) \\ \mathcal{O}\left(\epsilon^{-2}\right). & \text{if } b \in (1/2, 1] \end{cases} \tag{18}$$

Proof. See Appendix A □

The analysis of Lemma 2 and Theorem 1 do not make any assumptions on the size of the state action space. Additionally, the result describes the number of actor updates required. The number of critic updates required is simply the K_ϵ^{th} triangular number ,i.e. $\binom{K_\epsilon+1}{2}$.

4 Empirical Results

We numerically evaluate the rates at which actor-critic converges to a stationary point by observing the discounted reward and estimate of the gradient norm for a navigation problem. In particular,

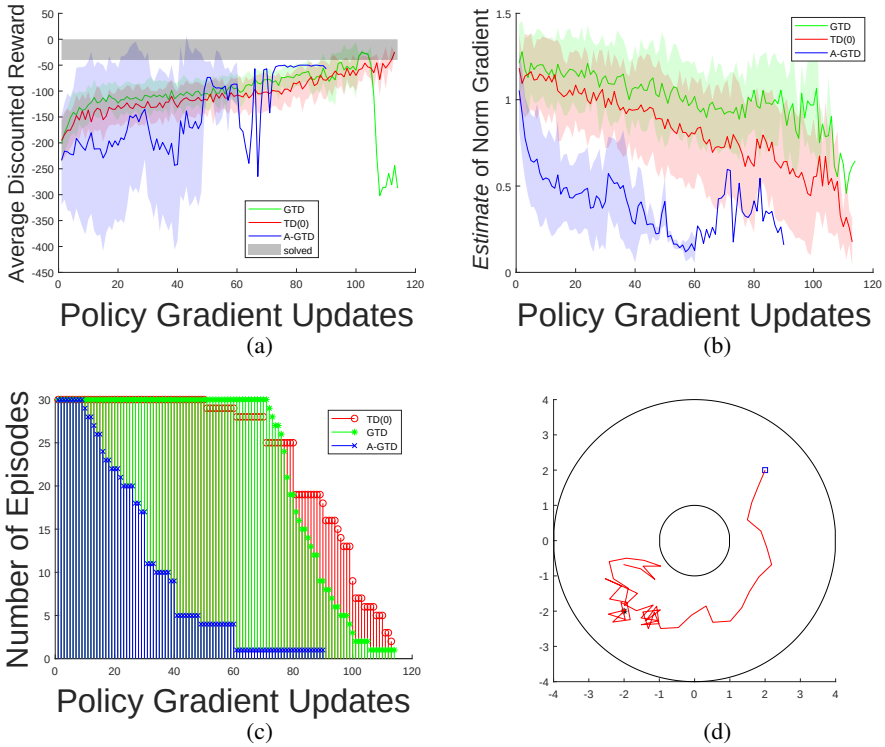

Figure 1: Convergence Results for actor-critic with TD(0), GTD, and A-GTD critic updates. (a) shows the average discounted reward. Average reward greater than -40 (shown in gray) is considered solved. (b) Estimate of the norm gradient (c) support of the first two plots (d) Example of a solved rollout with total reward equal to -35.6

we consider the problem where a point agent starts at an initial position $s_0 = (2, 2)$ and must learn to navigate to a region around a target $s_G = (-2, -2)$ while avoiding collisions with an obstacle and outer boundary. Given the state at time t, the action a_t is sampled from a multivariate Gaussian distribution parametrized by mean $\theta^\top \varphi(s_t)$ and covariance matrix $\Sigma = 0.5 \cdot I_2$. Given the action, the state transition is determined by $s_{t+1} = s_t + 0.5 a_t / \|a_t\|$. The state is represented by the agent's location in $\mathbb{R}^2$. A circular obstacle with radius one is centered at the origin, and the goal of the agent is a circular disc with radius 0.5 located at s_G. The agent is out of bounds if it steps outside the disc centered at the origin with radius equal to four. Wanting the agent to reach the target while avoiding the obstacle and outer boundary, the agent sees $r_t = -11$ if it hits an obstacle or boundary, $r_t = -1$ if it remains in the free space, and $r_t = -0.1$ if it hits the target region.

To evaluate the policy, we take the average reward of ten trajectories with length 50. This means that average rewards greater than -40 are avoiding obstacles while reaching the destination and are therefore considered solved (see Figure 1 (d)). To estimate the norm of the gradient squared, we take the average of the gradient estimate at the end ten trajectories with lengths determined by sampling a geometric distribution with parameter $1 - \gamma$ (discount factor $\gamma = 0.9$). The feature representation of the state is determined by a radial basis (Gaussian) kernel grid where $\kappa(x, x') = \exp\left\{-\|x - x'\|_2^2 / 2\sigma^2\right\}$. The 100 grid points are chosen evenly between -5 and 5 on both axes with $\sigma^2 = 0.1$. For all simulations, we chose a constant actor stepsize $\eta_k = 0.001$.

We considered actor-critic with classic Temporal Difference (TD(0)), Gradient Temporal Difference (GTD), and Accelerated Gradient Temporal Difference (A-GTD) critic updates. The algorithm was implemented for 30 episodes, terminated early if the average reward was less greater than -40, the estimate of the norm gradient was less than 0.2, or the actor parameter's norm exceeded 25. Figure 1 characterizes the rate with confidence intervals shaded show the standard deviation. Unsurprisingly, for all critic methods, the increase in reward corresponds to a decrease in the gradient norm. A-GTD converges very quickly, but its mean never reaches the *solved* region denoted by the gray shade. Indeed, these rates characterize corroborate Theorem 1, showing that actor-critic complexity is bottlenecked by the critic when the critic has a slow convergence rate.

References

[1] John Asmuth and Michael L Littman. Approaching bayes-optimalilty using monte-carlo tree search. In *Proc. 21st Int. Conf. Automat. Plan. Sched., Freiburg, Germany*, 2011.

[2] Richard Bellman. The theory of dynamic programming. Technical report, RAND Corp Santa Monica CA, 1954.

[3] Dimitri P Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. 2005.

[4] Jalaj Bhandari, Daniel Russo, and Raghav Singal. A finite time analysis of temporal difference learning with linear function approximation. *arXiv preprint arXiv:1806.02450*, 2018.

[5] Shalabh Bhatnagar, Mohammad Ghavamzadeh, Mark Lee, and Richard S Sutton. Incremental natural actor-critic algorithms. In *Advances in Neural Information Processing Systems*, pages 105–112, 2008.

[6] Shalabh Bhatnagar, Richard Sutton, Mohammad Ghavamzadeh, and Mark Lee. Natural actor-critic algorithms. *Automatica*, 45(11):2471–2482, 2009.

[7] Vivek S Borkar. Stochastic approximation with two time scales. *Systems & Control Letters*, 29(5):291–294, 1997.

[8] Vivek S Borkar. *Stochastic approximation: a dynamical systems viewpoint*, volume 48. Springer, 2009.

[9] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, Schulman John, Tang Jie, and Zaremba Wojciech. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[10] Dotan Di Castro and Ron Meir. A convergent online single-time-scale actor-critic algorithm. *Journal of Machine Learning Research*, 11(Jan):367–410, 2010.

[11] Gal Dalal, Balazs Szorenyi, Gugan Thoppe, and Shie Mannor. Finite sample analysis of two-timescale stochastic approximation with applications to reinforcement learning. *arXiv preprint arXiv:1703.05376*, 2017.

[12] Kenji Doya. Reinforcement learning in continuous time and space. *Neural Computation*, 12(1):219–245, 2000.

[13] Sylvain Gelly and David Silver. Monte-carlo tree search and rapid action value estimation in computer go. *Artificial Intelligence*, 175(11):1856–1875, 2011.

[14] Ilaria Giannoccaro and Pierpaolo Pontrandolfo. Inventory management in supply chains: a reinforcement learning approach. *International Journal of Production Economics*, 78(2):153–161, 2002.

[15] Daniel R Jiang, Thuy V Pham, Warren B Powell, Daniel F Salas, and Warren R Scott. A comparison of approximate dynamic programming techniques on benchmark energy storage problems: Does anything work? In *2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 1–8. IEEE, 2014.

[16] Jens Kober and Jan Peters. Reinforcement learning in robotics: A survey. In *Reinforcement Learning*, pages 579–610. Springer, 2012.

[17] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in Neural Information Processing Systems*, pages 1008–1014, 2000.

[18] Vijaymohan R Konda and Vivek S Borkar. Actor-critic–type learning algorithms for Markov decision processes. *SIAM Journal on Control and Optimization*, 38(1):94–123, 1999.

[19] Harold J. Kushner and G. George Yin. *Stochastic approximation and recursive algorithms and applications*. Springer, New York, NY, 2003.

[20] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937, 2016.

[21] Matteo Papini, Damiano Binaghi, Giuseppe Canonaco, Matteo Pirotta, and Marcello Restelli. Stochastic variance-reduced policy gradient. In *International Conference on Machine Learning*, pages 4026–4035, 2018.

[22] Santiago Paternain. *Stochastic Control Foundations of Autonomous Behavior*. PhD thesis, University of Pennsylvania, 2018.

[23] Matteo Pirotta, Marcello Restelli, and Luca Bascetta. Policy gradient in Lipschitz Markov Decision processes. *Machine Learning*, 100(2-3):255–283, 2015.

[24] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[25] Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.

[26] Richard S Sutton, Andrew G Barto, et al. *Reinforcement Learning: An Introduction*. 2 edition, 2017.

[27] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, pages 1057–1063, 2000.

[28] Richard S Sutton, Csaba Szepesvári, and Hamid Reza Maei. A convergent o (n) algorithm for off-policy temporal-difference learning with linear function approximation. *Advances in neural information processing systems*, 21(21):1609–1616, 2008.

[29] Gerald Tesauro. Temporal difference learning and td-gammon. 1995.

[30] John N Tsitsiklis. Asynchronous stochastic approximation and q-learning. *Machine learning*, 16(3):185–202, 1994.

[31] John N Tsitsiklis and Benjamin Van Roy. Analysis of temporal-diffference learning with function approximation. In *Advances in Neural Information Processing Systems*, pages 1075–1081, 1997.

[32] Xiao Wang, Shiqian Ma, Donald Goldfarb, and Wei Liu. Stochastic quasi-newton methods for nonconvex stochastic optimization. *SIAM Journal on Optimization*, 27(2):927–956, 2017.

[33] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[34] Kaiqing Zhang, Alec Koppel, Hao Zhu, and Tamer Başar. Convergence and iteration complexity of policy gradient method for infinite-horizon reinforcement learning. *IEEE Conference on Decision and Control (to appear)*, Dec. 2019.

[35] Kaiqing Zhang, Alec Koppel, Hao Zhu, and Tamer Başar. Global convergence of policy gradient methods: A nonconvex optimization perspective. *SIAM Journal on Control and Optimization (SICON) (submitted)*, Jan 2019.

[36] Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Başar. Fully decentralized multi-agent reinforcement learning with networked agents. In *International Conference on Machine Learning*, pages 5872–5881, 2018.

[37] Shaofeng Zou, Tengyu Xu, and Yingbin Liang. Finite-sample analysis for sarsa and q-learning with linear function approximation. *arXiv preprint arXiv:1902.02234*, 2019.

A Proof of Theorem 1

We begin by recalling [35][Lemma 4.2]

Lemma 1 (Lipschitz-Continuity of Policy Gradient)**.** *The policy gradient $\nabla J(\theta)$ is Lipschitz continuous with some constant $L > 0$, i.e., for any $\theta^1, \theta^2 \in \mathbb{R}^d$*

$$\|\nabla J(\theta^1) - \nabla J(\theta^2)\| \leq L \cdot \|\theta^1 - \theta^2\|. \tag{19}$$

This lemma allows us to establish an approximate ascent lemma for a random variable W_k defined by

$$W_k = J(\theta_k) - L\sigma^2 \sum_{j=k}^{\infty} \eta_j^2, \tag{20}$$

where $J(\theta)$ is defined in (2), σ^2 is defined in Assumption 8, and L is the Lipshitz constant of the gradient from Lemma 1. Unless otherwise stated, to alleviate notation, we denote ξ_k as short-hand for $\xi_{T_C(k)}$.

Lemma 2. Consider the actor parameter sequence defined by Algorithm 1. The sequence $\{W_k\}$ defined in (20) satisfies the inequality

$$\mathbb{E}[W_{k+1}|\mathcal{F}_k] \geq W_k + \eta_k \|\nabla J(\theta_k)\|^2 - \eta_k C \mathbb{E}[\|\xi_k - \xi_*\||\mathcal{F}] \tag{21}$$

where $C = B_\Theta C_2 C_3/(1-\gamma)$, with B_Θ the bound on the score function as in Assumption 1, C_2 the bound on the feature map in Assumption 2, and C_3 as the bound on the value function in (7).

Proof. See AppendixB □

Given the aforementioned lemmas, we are now in the position to begin the main argument. Begin by substituting the definition for W_k [cf. (20)] into Lemma 2, i.e., (21) to write

$$\mathbb{E}[J(\theta_{k+1})|\mathcal{F}_k] - L\sigma^2 \sum_{j=k+1}^{\infty} \eta_j^2 \geq J(\theta_k) - L\sigma^2 \sum_{j=k}^{\infty} \eta_j^2 + \eta_k \|\nabla J(\theta_k)\|^2 - \eta_k C \mathbb{E}[\|\xi_k - \xi_*\||\mathcal{F}_k]. \tag{22}$$

Take the total expectation and the term $\sum_{j=k+1}^{\infty} \eta_j^2$ cancels from both sides

$$\mathbb{E}[J(\theta_{k+1})] \geq \mathbb{E}[J(\theta_k)] - L\sigma^2 \eta_k^2 + \eta_k \|\nabla J(\theta_k)\|^2 - \eta_k C \mathbb{E}[\|\xi_k - \xi_*\|]. \tag{23}$$

Define $U_k := J(\theta^*) - J(\theta_k)$ where θ^* is the solution of (2) when the policy is parametrized by θ. By this definition, we know that U_k is non-negative for all θ_k. Add $J(\theta^*)$ to both sides of the inequality and rearrange terms

$$\eta_k \mathbb{E}[\|\nabla J(\theta_k)\|] \leq \mathbb{E}[U_k] - \mathbb{E}[U_{k+1}] + L\sigma^2 \eta_k^2 + \eta_k C \mathbb{E}[\|\xi_k - \xi_*\|]. \tag{24}$$

Divide both sides by η_k and take the sum over $\{k-N, \ldots, k\}$ where $0 < N \in \mathbb{Z}$

$$\sum_{j=k-N}^{k} \mathbb{E}[\|\nabla J(\theta_j)\|^2] \leq \sum_{j=k-N}^{k} \frac{1}{\eta_j} \left(\mathbb{E}[U_j] - \mathbb{E}[U_{j+1}]\right) + L\sigma^2 \sum_{j=k-N}^{k} \eta_j + C \sum_{j=k-N}^{k} \mathbb{E}[\|\xi_j - \xi_*\|]. \tag{25}$$

Add and subtract $1/\eta_{k-N-1}\mathbb{E}[U_{k-N}]$ on the right hand side. This allows us to write

$$\begin{aligned}\sum_{j=k-N}^{k} \mathbb{E}[\|\nabla J(\theta_j)\|^2] \leq \sum_{j=k-N}^{k} \left(\frac{1}{\eta_j} - \frac{1}{\eta_{j-1}}\right) \mathbb{E}[U_j] - \frac{1}{\eta_k}\mathbb{E}[U_{k+1}] + \frac{1}{\eta_{k-N-1}}\mathbb{E}[U_{k-N}] \\ + L\sigma^2 \sum_{j=k-N}^{k} \eta_j + C \sum_{j=k-N}^{k} \mathbb{E}[\|\xi_j - \xi_*\|].\end{aligned} \tag{26}$$

By definition of U_k, $\mathbb{E}[U_{k+1}] \geq 0$. Therefore we can omit it from the right hand side of (26). Further, we know that $J(\theta^*) \leq U_R$ by Assumption 1(i). Hence we have $U_k \leq 2U_R/(1-\gamma) =: C_4$ for all k.

Substituting this fact into the preceding expression yields

$$\sum_{j=k-N}^{k} \mathbb{E}[\|\nabla J(\theta_j)\|^2] \leq \sum_{j=k-N}^{k} \left(\frac{1}{\eta_j} - \frac{1}{\eta_{j-1}}\right) C_4 + \frac{1}{\eta_{k-N-1}} C_4 + L\sigma^2 \sum_{j=k-N}^{k} \eta_j + C \sum_{j=k-N}^{k} \mathbb{E}[\|\xi_j - \xi_*\|]. \tag{27}$$

By unraveling the telescoping sum, the first two terms are equal to C_4/η_k

$$\sum_{j=k-N}^{k} \mathbb{E}[\|\nabla J(\theta_j)\|^2] \leq \frac{C_4}{\eta_k} + L\sigma^2 \sum_{j=k-N}^{k} \eta_j + C \sum_{j=k-N}^{k} \mathbb{E}[\|\xi_j - \xi_*\|]. \tag{28}$$

Substitute the $\eta_k = k^{-a}$ for the step size and apply the bound (14) in Assumption 4 using extended notation $\xi_k = \xi_{T_C(k)}$

$$\sum_{j=k-N}^{k} \mathbb{E}[\|\nabla J(\theta_j)\|^2] \leq C_4 k^a + L\sigma^2 \sum_{j=k-N}^{k} j^{-a} + CL_1 \sum_{j=k-N}^{k} T_C(j)^{-b}. \tag{29}$$

We break the remainder of the proof into two cases due to the fact that the right-hand side of the preceding expression simplifies when $b = 1$, and is more intricate when $0 < b < 1$. We focus on the later case first.

Case (i): $b \in (0, 1)$ Consider the case where $b \in (0, 1)$. Set $T_C(k) = k$. Substitute the integration rule, namely that $\sum_{j=k-N}^{k} j^{-a} \leq k^{1-a} - (k - N - 1)^{1-a}$, into (29) to obtain:

$$\sum_{j=k-N}^{k} \mathbb{E}[\|\nabla J(\theta_j)\|^2] \leq C_4 k^a + \frac{L\sigma^2}{1-a}\left(k^{1-a} - (k-N-1)^{1-a}\right) + \frac{CL_1}{1-b}\left(k^{1-b} - (k-N-1)^{1-b}\right). \tag{30}$$

Divide both sides by k and set $N = k - 1$

$$\frac{1}{k}\sum_{j=1}^{k} \mathbb{E}[\|\nabla J(\theta_j)\|^2] \leq C_4 k^{a-1} + \frac{L\sigma^2}{1-a} k^{-a} + \frac{CL_1}{1-b} k^{-b}. \tag{31}$$

Suppose $k = K_\epsilon$ so that we may write

$$\frac{1}{K_\epsilon}\sum_{j=1}^{K_\epsilon} \mathbb{E}[\|\nabla J(\theta_j)\|^2] \leq \mathcal{O}\left(K_\epsilon^{a-1} + K_\epsilon^{-a} + K_\epsilon^{-b}\right). \tag{32}$$

By definition of K_ϵ, we have

$$\epsilon \leq \frac{1}{K_\epsilon}\sum_{j=1}^{K_\epsilon} \mathbb{E}[\|\nabla J(\theta_j)\|^2] \leq \mathcal{O}\left(K_\epsilon^{a-1} + K_\epsilon^{-a} + K_\epsilon^{-b}\right). \tag{33}$$

Defining $\ell = \min\{a, 1-a, b\}$, the preceding expression then implies

$$\epsilon \leq \mathcal{O}(K_\epsilon^{-\ell}), \tag{34}$$

which by inverting the expression, yields the sample complexity

$$K_\epsilon \leq \mathcal{O}(\epsilon^{-1/\ell}). \tag{35}$$

Case (ii): $b = 1$ Now consider the case where $b = 1$. Set $T_C(k) = k + 1$. Again, using the integration rule, and that $\sum_{j=k-N}^{k} (j+1)^{-1} \leq \log(k+1) - \log(k-N)$, we substitute into (29) which yields

$$\sum_{j=k-N}^{k} \mathbb{E}[\|\nabla J(\theta_j)\|^2] \leq C_4 k^a + \frac{L\sigma^2}{1-a}\left(k^{1-a} - (k-N-1)^{1-a}\right) + CL_1\left(\log(k+1) - \log(k-N)\right). \tag{36}$$

Divide both sides by k and fix $N = k - 1$

$$\frac{1}{k}\sum_{j=1}^{k}\mathbb{E}[\|\nabla J(\theta_j)\|^2] \leq C_4 k^{a-1} + \frac{L\sigma^2}{1-a}k^{-a} + CL_1\frac{\log(k+1)}{k}. \tag{37}$$

Let $k = K_\epsilon$ in the preceding expression, which then becomes

$$\frac{1}{K_\epsilon}\sum_{j=1}^{K_\epsilon}\mathbb{E}[\|\nabla J(\theta_j)\|^2] \leq \mathcal{O}\left(K_\epsilon^{a-1} + K_\epsilon^{-a} + \frac{\log(K_\epsilon+1)}{K_\epsilon}\right). \tag{38}$$

By definition of K_ϵ, we have

$$\epsilon \leq \frac{1}{K_\epsilon}\sum_{j=1}^{K_\epsilon}\mathbb{E}[\|\nabla J(\theta_j)\|^2] \leq \mathcal{O}\left(K_\epsilon^{a-1} + K_\epsilon^{-a} + \frac{\log(K_\epsilon+1)}{K_\epsilon}\right). \tag{39}$$

Optimizing over a, we have

$$\epsilon \leq \mathcal{O}\left(K_\epsilon^{-\frac{1}{2}}\right) \quad \text{for } b > \frac{1}{2} \tag{40}$$

On the other hand,

$$\epsilon \leq \mathcal{O}\left(K_\epsilon^{-b}\right). \text{ for } b \leq 1/2 \tag{41}$$

Fix $\ell = \min\{1/2, b\}$, then

$$\epsilon \leq \mathcal{O}(K_\epsilon^{-\ell}), \tag{42}$$

which implies

$$K_\epsilon \leq \mathcal{O}(\epsilon^{-1/\ell}). \tag{43}$$

This concludes the proof.

B Proof of Lemma 2

By definition of W_k, we write the expression for W_{k+1}

$$W_{k+1} = J(\theta_{k+1}) - L\sigma^2\sum_{j=k+1}^{\infty}\eta_j^2. \tag{44}$$

By the Mean Value Theorem, there exists $\tilde{\theta}_k = \lambda\theta_k + (1-\lambda)\theta_{k+1}$ for some $\lambda \in [0,1]$ such that

$$J(\theta_{k+1}) = J(\theta_k) + (\theta_{k+1} - \theta_k)^\top \nabla J(\tilde{\theta}_k). \tag{45}$$

Substitute this expression for $J(\theta_{k+1})$ in (44)

$$W_{k+1} = J(\theta_k) + (\theta_{k+1} - \theta_k)^\top \nabla J(\tilde{\theta}_k) - L\sigma^2\sum_{j=k+1}^{\infty}\eta_j^2. \tag{46}$$

Add and subtract $(\theta_{k+1} - \theta_k)^\top \nabla J(\theta_k)$ to the right hand side of (46) to obtain

$$W_{k+1} = J(\theta_k) + (\theta_{k+1} - \theta_k)^\top \left(\nabla J(\tilde{\theta}_k) - \nabla J(\theta_k)\right) + (\theta_{k+1} - \theta_k)^\top \nabla J(\theta_k) - L\sigma^2\sum_{j=k+1}^{\infty}\eta_j^2. \tag{47}$$

By Cauchy Schwartz, we know $(\theta_{k+1} - \theta_k)^\top \left(\nabla J(\tilde{\theta}_k) - \nabla J(\theta_k)\right) \geq -\|\theta_{k+1} - \theta_k\|\|\nabla J(\tilde{\theta}_k) - J(\theta_k)\|$. Further, by the Lipschitz continuity of the gradient, we know $\|\nabla J(\tilde{\theta}_k) - \nabla J(\theta_k)\| \leq L\|\tilde{\theta}_k - \theta_k\|$. Therefore, we have

$$(\theta_{k+1} - \theta_k)^\top \left(\nabla J(\tilde{\theta}_k) - J(\theta_k)\right) \geq -L\|\tilde{\theta}_k - \theta_k\| \cdot \|\theta_{k+1} - \theta_k\| \geq -L\|\theta_{k+1} - \theta_k\|^2, \tag{48}$$

where the second inequality comes from substituting $\tilde{\theta}_k = (1-\lambda)\theta_{k+1} + \lambda\theta_k$. We substitute this expression into the definition of W_{k+1} in (47) to obtain

$$W_{k+1} \geq J(\theta_k) + (\theta_{k+1} - \theta_k)^\top \nabla J(\theta_k) - L\|\theta_{k+1} - \theta_k\|^2 - L\sigma^2\sum_{j=k+1}^{\infty}\eta_j^2. \tag{49}$$

Take the expectation with respect to the filtration $\mathcal{F}_k$, substitute the definition for the actor update (15), together with the fact that this update has bounded variance (8) (Assumption 3), to obtain

$$\mathbb{E}[W_{k+1}|\mathcal{F}_k] \geq J(\theta_k) + \mathbb{E}[\theta_{k+1} - \theta_k|\mathcal{F}_k]^\top \nabla J(\theta_k) - L\sigma^2\eta_k^2 - L\sigma^2 \sum_{j=k+1}^{\infty} \eta_j^2. \tag{50}$$

The terms on the right hand side outside the expectation may be identified as W_k [cf. (20)] by definition, which allows us to write

$$\mathbb{E}[W_{k+1}|\mathcal{F}_k] \geq W_k + \mathbb{E}[\theta_{k+1} - \theta_k|\mathcal{F}_k]^\top \nabla J(\theta_k). \tag{51}$$

Therefore, we are left to show that the last term on the right-hand side of the preceding expression is "nearly" an ascent direction, i.e.,

$$\mathbb{E}[\theta_{k+1} - \theta_k|\mathcal{F}_k]^\top \nabla J(\theta_k) \geq \eta_k \|\nabla J(\theta_k)\|^2 - \eta_k C \mathbb{E}[\|\xi_k - \xi_*\|^2|\mathcal{F}_k]. \tag{52}$$

and how far from an ascent direction it is depends on the critic estimate bias $\mathbb{E}[\|\xi_k - \xi_*\|^2|\mathcal{F}_k]$. From Algorithm 1, the actor parameter update may be written as

$$\theta_{k+1} - \theta_k = \frac{1}{1-\gamma}\eta_k \xi_k^\top \varphi(s_{T_k}, a_{T_k}) \nabla \log \pi(s_{T_k}, a_{T_k}|\theta_k). \tag{53}$$

Add and subtract $\eta_k \xi_*^\top \varphi(s_{T_k}, a_{T_k}) \nabla \log \pi(s_{T_k}, a_{T_k}|\theta_k)$ to (53) where ξ_* is such that the estimate $\hat{Q}_{\pi_{\theta_k}}(s,a) = \xi_*^\top \varphi(s,a) = Q_{\pi_{\theta_k}}(s,a)$ is unbiased. Hence, $\|\xi_{T_k} - \xi_*\|$ represents the distance between the critic parameters corresponding to the biased estimate after k critic only steps and the true estimate of the Q function.

$$\begin{aligned}\theta_{k+1} - \theta_k =& \frac{1}{1-\gamma}\eta_k (\xi_k - \xi_*)^\top \varphi(s_{T_k}, a_{T_k}) \nabla \log \pi(s_{T_k}, a_{T_k}|\theta_k) \\ &+ \frac{1}{1-\gamma}\eta_k \xi_*^\top \varphi(s_{T_k}, a_{T_k}) \nabla \log \pi(s_{T_k}, a_{T_k}|\theta_k).\end{aligned} \tag{54}$$

Here we recall (11) and (12) from the derivation of the algorithm, that is that the expected value of the stochastic estimate given θ is unbiased. Therefore, by taking the expectation of (54) with respect to the filtration $\mathcal{F}_k$, we obtain

$$\mathbb{E}[\theta_{k+1} - \theta_k|\mathcal{F}_k] = \frac{1}{1-\gamma}\eta_k \mathbb{E}[(\xi_k - \xi_*)^\top \varphi(s_{T_k}, a_{T_k}) \nabla \log \pi(s_{T_k}, a_{T_k}|\theta_k)|\mathcal{F}_k] + \eta_k \nabla J(\theta_k). \tag{55}$$

Take the inner product with $\nabla J(\theta_k)$ on both sides

$$\begin{aligned}\mathbb{E}[\theta_{k+1} - \theta_k|\mathcal{F}_k]^\top \nabla J(\theta_k) =& \frac{1}{1-\gamma}\eta_k \mathbb{E}[(\xi_k - \xi_*)^\top \varphi(s_{T_k}, a_{T_k}) \nabla \log \pi(s_{T_k}, a_{T_k}|\theta_k)|\mathcal{F}_k]^\top \nabla J(\theta_k) \\ &+ \eta_k \|\nabla J(\theta_k)\|^2\end{aligned} \tag{56}$$

The first term on the right-hand side is lower-bounded by the negative of its absolute value, i.e.,

$$\begin{aligned}&\mathbb{E}[\theta_{k+1} - \theta_k|\mathcal{F}_k]^\top \nabla J(\theta_k) \geq \\ &- \frac{1}{1-\gamma}\eta_k |\mathbb{E}[(\xi_k - \xi_*)^\top \varphi(s_{T_k}, a_{T_k}) \nabla \log \pi(s_{T_k}, a_{T_k}|\theta_k)|\mathcal{F}_k]^\top \nabla J(\theta_k)| + \eta_k \|\nabla J(\theta_k)\|^2.\end{aligned} \tag{57}$$

Next, we apply Cauchy Schwartz to the first term on the right-hand side of the previous expression, followed by Jensen's Inequality, and then Cauchy Schwartz again, to obtain

$$\begin{aligned}&\mathbb{E}[\theta_{k+1} - \theta_k|\mathcal{F}_k]^\top \nabla J(\theta_k) \\ &\geq -\frac{1}{1-\gamma}\eta_k \|\mathbb{E}[(\xi_k - \xi_*)^\top \varphi(s_{T_k}, a_{T_k}) \nabla \log \pi(s_{T_k}, a_{T_k}|\theta_k)|\ |\mathcal{F}_k]\| \cdot \|\nabla J(\theta_k)\| + \eta_k \|\nabla J(\theta_k)\|^2 \\ &\geq -\frac{1}{1-\gamma}\eta_k \mathbb{E}[|(\xi_k - \xi_*)^\top \varphi(s_{T_k}, a_{T_k})| \|\nabla \log \pi(s_{T_k}, a_{T_k}|\theta_k)\|| \ |\mathcal{F}_k] \cdot \|\nabla J(\theta_k)\| + \eta_k \|\nabla J(\theta_k)\|^2 \\ &\geq -\frac{1}{1-\gamma}\eta_k \mathbb{E}[\|(\xi_k - \xi_*)\| \cdot \|\varphi(s_{T_k}, a_{T_k})\| \cdot \|\nabla \log \pi(s_{T_k}, a_{T_k}|\theta_k)\|| \ |\mathcal{F}_k] \cdot \|\nabla J(\theta_k)\| \\ &\quad + \eta_k \|\nabla J(\theta_k)\|^2.\end{aligned} \tag{58}$$

Because the score function ($\|\nabla \log \pi(s,a|\theta)\| \leq B_\Theta$), the feature map ($\|\varphi(s,a)\| \leq C_2$), and the gradient ($\|\nabla J(\theta_k)\| \leq C_3$) are bounded, we define C be the product of these constants with $1/(1-\gamma)$:

$$\|\varphi(s_{T_k}, a_{T_k})\| \cdot \|\nabla \log \pi(s_{T_k}, a_{T_k}|\theta_k)\| \cdot \|\nabla J(\theta_k)\| \leq \frac{B_\Theta C_2 C_3}{1-\gamma} =: C. \tag{59}$$

which my be substituted into (58) to write

$$\mathbb{E}[\theta_{k+1} - \theta_k | \mathcal{F}_k]^\top \nabla J(\theta_k) \geq -C\eta_k \mathbb{E}[\|\xi_k - \xi_*\| | \mathcal{F}_k] + \eta_k \|\nabla J(\theta_k)\|^2 \tag{60}$$

Now, we can express this relationship in terms of W_k by substituting back into (51):

$$\mathbb{E}[W_{k+1} | \mathcal{F}_k] \geq W_k - C\eta_k \mathbb{E}[\|\xi_k - \xi_*\| | \mathcal{F}_k] + \eta_k \|\nabla J(\theta_k)\|^2 \tag{61}$$

which is as stated in (21).

C Extended Discussion

Our main result states that actor critic converges at rate $\mathcal{O}(\epsilon^{-2})$ when the critic converges at a rate faster than $k^{-1/2}$ and at rate $\mathcal{O}(\epsilon^{-1/b})$ when the critic converges slower than $k^{-1/2}$. This means that when the actor critic algorithm uses a critic method which converges quickly, the actor critic algorithm will converge faster than otherwise. Consider the critic convergence rates characterized by Temporal Difference (TD(0)), Gradient Temporal Difference (GTD), and Accelerated Gradient Temporal Difference (A-GTD). Depending on the assumptions on the state action space, TD(0) has been shown to have a convergence rate of $k^{-1/4}$ for continuous state action spaces. Using stochastic compositional gradient descent, GTD can be shown to have a convergence rate of $k^{-1/3}$ while A-GTD has the fastest convergence rate $k^{-2/5}$.

Indeed, our numerical simulations show that A-GTD, the actor-critic algorithm with the fastest critic only convergence rate, converges faster than GTD or TD(0). What is interesting, however is that the stationary point to which it converges is not considered *solved*. This means the agent only learned how to avoid obstacles, but did not learn to navigate to the target. Recall that each episode was terminated early if either the problem was solved, the norm of the gradient was below a certain threshold, or the norm of the actor parameter grew too large. Of the three algorithms, GTD performed the poorest, and was also the only algorithm which was terminated early due to the large actor parameter. We believe this was because of a poor choice of actor stepsize.

Taken together, these theoretical and experimental results suggest a tight coupling between the choice of training methodology and the quality of learned policies. Thus, just as the choice of optimization method, statistical model, and sample size influence generalization in supervised learning, they do so in reinforcement learning.

There are a number of future directions to take this work. To begin, we can explicitly characterize the rate of convergence for TD(0), GTD, and A-GTD by applying bounds of the form (14) from existing literature. Second, we can establish bounds on cases where the samples are not i.i.d., but instead have Markovian noise. Finally, we can characterize the behavior of the variance and employ such characterizations to accelerate training.