Towards Global Optimality for Practical Average Reward Reinforcement Learning without Mixing Time Oracles

Bhrij Patel [1] **Wesley A. Suttle** [2] **Alec Koppel** [3] **Vaneet Aggarwal** [4] **Brian M. Sadler** [5] **Dinesh Manocha** [1]
Amrit Singh Bedi [6]

Abstract

In the context of average-reward reinforcement learning, the requirement for oracle knowledge of the mixing time, a measure of the duration a Markov chain under a fixed policy needs to achieve its stationary distribution, poses a significant challenge for the global convergence of policy gradient methods. This requirement is particularly problematic due to the difficulty and expense of estimating mixing time in environments with large state spaces, leading to the necessity of impractically long trajectories for effective gradient estimation in practical applications. To address this limitation, we consider the Multi-level Actor-Critic (MAC) framework, which incorporates a Multi-level Monte-Carlo (MLMC) gradient estimator. With our approach, we effectively alleviate the dependency on mixing time knowledge, a first for average-reward MDPs global convergence. Furthermore, our approach exhibits the tightest available dependence of $\mathcal{O}\left(\sqrt{\tau_{mix}}\right)$ known from prior work. With a 2D grid world goal-reaching navigation experiment, we demonstrate that MAC outperforms the existing state-of-the-art policy gradient-based method for average reward settings.

1. Introduction

In reinforcement learning (RL) problems, temporal dependence of data breaks the independent and identically distributed (i.i.d.) assumption commonly encountered in machine learning analyses, rendering the theoretical analysis of RL methods challenging. In discounted RL, the impact of temporal dependence is typically mitigated, as the effect of the discount factor renders the stationary behavior of the induced Markov chains irrelevant. On the other hand, in average-reward RL, stationary behavior under induced policies is of fundamental importance. In particular, understanding the effect of mixing time, a measure of how long a Markov chain takes to approach stationarity, is critical to the development and analysis of average-reward RL methods (Suttle et al., 2023; Riemer et al., 2021). Given the usefulness of the average-reward regime in applications such as robotic locomotion (Zhang & Ross, 2021), traffic engineering (Geng et al., 2020), and healthcare (Ling et al., 2023), improving our understanding of the issues inherent in average-reward RL is increasingly important.

Key to theoretically understanding a learning method is characterizing its convergence behavior. For a method to be considered sound, we should ideally be able to prove that, under suitable conditions, it converges to a globally optimal solution while remaining sample-efficient. Convergence to global optimality of policy gradient (PG) methods (Sutton & Barto, 2018), a subset of RL methods well-suited to problems with large and complex state and action spaces, has been extensively studied in the discounted setting (Bhandari & Russo, 2024; Liu et al., 2020; Bedi et al., 2022; Gaur et al., 2023; Mondal & Aggarwal, 2024; Gaur et al., 2024). Due to gradient estimation issues arising from the mixing time dependence inherent in the average-reward setting, however, the problem of obtaining global optimality results for average-reward PG methods remained open until recently.

In Bai et al. (2024), the Parameterized Policy Gradient with Advantage Estimation (PPGAE) method was proposed and shown to converge to a globally optimal solution in average-reward problems under suitable conditions. However, the implementation of the PPGAE algorithm relies on oracle knowledge of mixing times, which are typically unknown and costly to estimate, and requires extremely long trajectory lengths at each gradient estimation step. These drawbacks render PPGAE costly and sample-inefficient, leav-

[1]Department of Computer Science, University of Maryland, College Park, USA. [2]US Army Research Laboratory, Adelphi, MD, USA. [3]JP Morgan AI Research, New York, USA. [4]School of Industrial Engineering, Purdue University, Indiana, USA. [5]Department of Computer Sicence, University of Texas, Austin, USA. [6]Department of Computer Science, University of Central Florida, Florida, USA.. Correspondence to: Bhrij Patel <bbp13@umd.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

ing open the problem of developing a practical average-reward PG method that enjoys global optimality guarantees. Recently, Suttle et al. (2023) proposed and analyzed the Multi-level Actor-critic (MAC) algorithm, an average-reward PG method that enjoys state-of-the-art sample complexity, avoids oracle knowledge of mixing times, and leverages a multi-level Monte Carlo (MLMC) gradient estimation scheme to keep trajectory lengths manageable. Despite these advantages, convergence to global optimality has not yet been provided for the MAC algorithm.

In this paper, we establish for the first time convergence to global optimality of an average-reward PG algorithm that does not require oracle knowledge of mixing times, uses practical trajectory lengths, and enjoys the best known dependence of convergence rate on mixing time. To achieve this, we extend the convergence analysis of (Bai et al., 2024) to the MAC algorithm of (Suttle et al., 2023), closing an outstanding gap in the theory of average-reward PG methods. In addition, we provide goal-reaching navigation results illustrating the superiority of MAC over PPGAE, lending further support to our theoretical contributions. We summarize our contributions as follows:

- We prove convergence of MAC to global optimality in the infinite horizon average-reward setting.
- Despite lack of mixing time knowledge, we achieve a tighter mixing time dependence, $\mathcal{O}\left(\sqrt{\tau_{mix}}\right)$, than previous average-reward PG algorithms.
- We highlight the practical feasibility of MAC compared with PPGAE by empirically comparing their sample complexities in a 2D gridworld goal-reaching navigation task where MAC achieves a higher reward.

The paper is organized as follows: in the next section we given an overview of related works in policy gradient algorithms and mixing time; Section 3 describes general problem formulation for average-reward policy gradient algorithms and then specifically details the MAC algorithm along with PPGAE from (Bai et al., 2024); Section 4 presents our global convergence guarantees of MAC and provides a discussion comparing the practicality of MAC to PPGAE. We also provide a 2D gridworld goal-reaching navigation experiment where MAC achieves a higher reward than PPGAE. We then end paper with conclusions and discussion of future work.

2. Related Works

In this section, we provide a brief overview of the related works for global optimality of policy gradient algorithms and for mixing time.

Policy Gradient. Convergence to global optimality of policy gradient methods has been established for softmax (Mei et al., 2020) and tabular (Bhandari & Russo, 2024; Agarwal et al., 2020) parameterizations. For the discounted reward setting, (Liu et al., 2020) provided a general framework for global optimality for PG and natural PG methods for the discounted reward setting and the optimal convergence rate guarantees have been studied in (Mondal & Aggarwal, 2024). Recently, Bai et al. (2024) adapted this framework for the average-reward infinite horizon MDP with general policy parameterization. In this work we apply this framework for the MAC algorithm to introduce a global optimality analysis in the average-reward setting with no oracle knowledge of mixing time.

Mixing Time. Previous works have emphasized the challenges and infeasibility of estimating mixing time (Hsu et al., 2015; Wolfer, 2020) in complex environments. Recently, Patel et al. (2023) used policy entropy as a proxy variable for mixing time for an adaptive trajectory length scheme. Previous works that assume oracle knowledge of mixing time such as Bai et al. (2024); Duchi et al. (2012); Nagaraj et al. (2020) are limited in practicality. In the Multi-level Actor-Critic (MAC) scheme proposed in Suttle et al. (2023), this assumption was relaxed while still recovering SOTA convergence by leveraging Multi-level Monte Carlo (MLMC) gradient estimation along the lines introduced by Dorfman & Levy (2022). In this paper, we establish the global convergence of MAC and highlight its tighter dependence on mixing time without requiring oracle knowledge of mixing time. We highlight the contributions and advancements of this paper compared to prior art (Bai et al., 2024; Dorfman & Levy, 2022; Suttle et al., 2023) in Table 2.

3. Problem Formulation

3.1. Average Reward Policy Optimization

The average-reward reinforcement learning problem may be formalized as a Markov decision process $\mathcal{M} := (\mathcal{S}, \mathcal{A}, \mathbb{P}, r)$, where $\mathcal{S}$ and $\mathcal{A}$ are the finite state and action spaces, respectively, $\mathbb{P}(\cdot \mid s, a)$ maps the current state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$ to the conditional probability distribution of next state $s' \in \mathcal{S}$, and $r : \mathcal{S} \times \mathcal{A} \to [0, r_{\max}]$ is a bounded reward function. An agent, starting from state $s_t \in \mathcal{S}$, selects actions $a_t \in \mathcal{A}$ which causes a transition to a new state $s_{t+1} \sim \mathbb{P}(\cdot \mid s_t, a_t)$, while the environment reveals a reward $r(s_t, a_t)$. Actions are selected according to a policy $\pi(\cdot \mid s)$, which is a distribution over action space $\mathcal{A}$ given current state s. The goal in this setting is to find a policy π maximizing the long-term average reward

$$J(\pi) := \lim_{T \to \infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T} r(s_t, a_t)\right].$$

We consider the parameterized setting this work, where the policy is parameterized by a vector $\theta \in \mathbb{R}^q$, where q denotes the parameter dimension, and the policy dependence on θ is

Table 1. This table compares the different policy gradient algorithms for the average reward setting and their global convergence rates. Out of all the papers with an explicit dependence on mixing time, MAC from (Suttle et al., 2023), which we analyze in this paper, has the tightest dependence.

Algorithm	Reference	Mixing Time Known	Mixing Time Dependence	Convergence Rate	Parameterization
FOPO	(Wei et al., 2021)	Yes	N/A	$\widetilde{\mathcal{O}}\left(T^{-\frac{1}{2}}\right)$	Linear
OLSVI.FH	(Wei et al., 2021)	Yes	N/A	$\widetilde{\mathcal{O}}\left(T^{-\frac{1}{4}}\right)$	Linear
MDP-EXP2	(Wei et al., 2021)	Yes	$\widetilde{\mathcal{O}}\left(\sqrt{\tau_{mix}^3}\right)$	$\widetilde{\mathcal{O}}\left(T^{-\frac{1}{2}}\right)$	Linear
PPGAE	(Bai et al., 2024)	Yes	$\widetilde{\mathcal{O}}\left(\tau_{mix}^2\right)$	$\widetilde{\mathcal{O}}\left(T^{-\frac{1}{4}}\right)$	General
MAC **(This work)**	(Suttle et al., 2023)	No	$\widetilde{\mathcal{O}}\left(\sqrt{\tau_{mix}}\right)$	$\widetilde{\mathcal{O}}\left(T^{-\frac{1}{4}}\right)$	General

Table 2. This table highlights the contributions of this work compared to prior work in global optimality (Bai et al., 2024) and mixing time (Dorfman & Levy, 2022; Suttle et al., 2023).

Reference	Global Optimality	General Policy Parameterization	Mixing Time Assumptions Removed	Practical Number of Samples
(Dorfman & Levy, 2022)	✗	✗	✓	✓
(Suttle et al., 2023)	✗	✓	✓	✓
(Bai et al., 2024)	✓	✓	✗	✗
This Work	✓	✓	✓	✓

indicated via the notation π_θ. Parameterization in practice can vary widely, from neural networks to tabular representations. This work, like in (Bai et al., 2024), aims to provide a global convergence guarantee with no assumption on policy parameterization. With this notation, we can formalize the objective as solving the following maximization problem:

$$\max_\theta J(\pi_\theta) := \lim_{T\to\infty} \mathbb{E}_{s_{t+1}\sim\mathbb{P}(\cdot|s_t,a_t),a_t\sim\pi_\theta(\cdot|s_t)}\left[R_T\right], \quad (1)$$

where $R_T := \frac{1}{T}\sum_{t=0}^{T} r(s_t,a_t)$. Observe that, in general, (1) is non-convex with respect to θ, which is the critical challenge of applying first-order iterations to solve this problem – see (Zhang et al., 2020; Agarwal et al., 2020). We furthermore define the stationary distribution induced by a given policy π_θ to be

$$d^{\pi_\theta}(s) = \lim_{T\to\infty}\frac{1}{T}\left[\sum_{t=0}^{T-1}\Pr(s_t=s|s_0\sim\rho,\pi_\theta)\right]. \quad (2)$$

As we will later discuss, the induced d^{π_θ} for a given θ is unique and is therefore agnostic to initial distribution ρ, due ergodicity assumptions. We are now able to express the average reward with respect to d^{π_θ} induced by a parameterized policy as $J(\pi_\theta) = \mathbb{E}_{s\sim d^{\pi_\theta},a\sim\pi_\theta}[r(s,a)]$. This equation reveals the explicit dependence our optimization problem has on d^{π_θ}. It is assumed that the data sampled comes from the unique stationary distribution. Any samples that are generated before the induced Markov Chain reaches a stationary distribution are known as burn-in samples and can lead to noisy gradients. Thus, knowing the number of timesteps it takes an induced Markov Chain to reach its stationary distribution is a crucial element for policy gradient algorithms. The formally capture this notion, the quantity is known as *mixing time* τ_{mix} is defined as follows.

Definition 1 (ϵ-Mixing Time)**.** Let d^{π_θ} denote the stationary distribution of the Markov chain P_θ induced by π_θ. Let

$$m(t;\theta) := \sup_{s\in\mathcal{S}} \|P_\theta^t(\cdot|s) - d^{\pi_\theta}(\cdot)\|_{TV}, \quad (3)$$

where $\|\cdot\|_{TV}$ is the total variation distance. The ϵ-mixing time of a Markov chain parameterized by θ is defined as

$$\tau_{mix}^\theta(\epsilon) := \inf\{t : m(t;\theta)\le\epsilon\}, \quad (4)$$

In PG and other analyses of mixing time, it is useful to define $\tau_{mix}^\theta := \tau_{mix}^\theta(1/4)$, as it implies the result that $m(l\tau_{mix}^\theta;\theta) \le 2^{-l}\}$ (Dorfman & Levy, 2022). Finally, given a timestep $T\in\mathbb{N}$, we define $\tau_{mix} = \max_{t\in[T]}\tau_{mix}^{\theta_t}$.

In prior works such as (Duchi et al., 2012; Nagaraj et al., 2020; Bai et al., 2024; Wei et al., 2021), τ_{mix}^θ is assumed to be known. However, for complex environments this is rarely the case (Hsu et al., 2015; Wolfer, 2020). In Section 3.4 we discuss the Multi-level Actor-Critic algorithm and how it relaxes this assumption. We will first briefly introduce the elements of a vanilla actor-critic in Section 3.3.

To do so, we define the action-value (Q) function as

$$Q^{\pi_\theta}(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty}[r(s_t,a_t) - J(\pi_\theta)]\right], \quad (5)$$

such that $s_0 = s, a_0 = a$, and action $a\sim\pi_\theta$. We can then further write the state value function as

$$V^{\pi_\theta}(s) = \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[Q^{\pi_\theta}(s,a)]. \quad (6)$$

Using Bellman's Equation and the definitions (5) and (6), we can write (Puterman, 2014)

$$V^{\pi_\theta}(s) = \mathbb{E}[r(s,a) - J(\pi_\theta) + V^{\pi_\theta}(s')], \tag{7}$$

where the expectation is over $a \sim \pi_\theta(\cdot|s), s' \sim \mathbb{P}(\cdot|a,s)$. We also define the advantage function by

$$A^{\pi_\theta}(s,a) \triangleq Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s). \tag{8}$$

With this, we can now present the well-known policy gradient theorem established by (Sutton et al., 1999),

Lemma 1. *The gradient of the long-term average reward can be expressed as follows.*

$$\nabla_\theta J(\theta) = \mathbf{E}_{s\sim d^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)}\Big[A^{\pi_\theta}(s,a)\nabla_\theta \log \pi_\theta(a|s)\Big]. \tag{9}$$

PG algorithms maximize the average reward by updating θ using a gradient ascent step with stepsize α [1],

$$\theta_{t+1} = \theta_t + \alpha_t \nabla_\theta J(\pi_{\theta_t}), \tag{10}$$

Normally, average-reward policy gradient algorithms estimate the advantage function with a simple average of all reward values observed during a trajectory. In (Bai et al., 2024) they propose an advantage estimation algorithm that estimates Q and V values from the trajectories sampled by splitting them into sub-trajectories. However, they propose dividing the trajectory into sub-trajectories, where the length of the length of trajectory and sub-trajectories are functions of mixing time. Thus, we aim to provide a global convergence guarantee of an algorithm that has no requirement of knowing mixing time. This goal motivates us to select MAC which has no such requirement but lacks global convergence analysis. We give an high-level overview of the algorithm from (Bai et al., 2024) in the following subsection.

3.2. Parameterized Policy Gradient with Advantage Estimation

Recently, (Bai et al., 2024) proposed a PG algorithm, Parameterized Policy Gradient with Advantage Estimation (PPGAE), and derived its global convergence gaurantee in the average reward setting for general policy parameterization. PPGAE defines K as the number of gradient updates, H as the length of trajectory, and T as the sample budget of the entire training process. Therefore, we can express them in terms of each with $K = T/H$. To formulate the policy gradient update at the end of the trajectory, the advantage value of each sample collected in a trajectory is then estimated based on the reward values observed in the samples. A more detailed description of the PPGAE algorithm can be found in the Appendix. In PPGAE, $H = 16\tau_{hit}\tau_{mix}\sqrt{T}(\log T)^2$, where τ_{hit}, *hitting time* is defined as below:

$$\tau_{hit} := \max_\theta \max_{s\in\mathcal{S}} \frac{1}{d^{\pi_\theta}(s)}. \tag{11}$$

[1]With slight abuse of notation, we use t as an index for sample number and gradient update to align with prior work in mixing time (Suttle et al., 2023; Dorfman & Levy, 2022).

Intuitively, it is the amount of time to reach all states in the state space. If the induced Markov Chain is ergodic, then τ_{hit} is finite because each state in $\mathcal{S}$ has a non-zero chance of being reached. Similar to mixing time, hitting time is also defined by the stationary distribution of a given policy. Thus, its estimation also suffers from the same difficulties as mixing time estimation.

The algorithm relies on knowing the mixing time and hitting time to calculate H, restricting its use case to simple environments with small state spaces where they can be feasibly be estimated. This requirement becomes more impractical as the state space or environment complexity increases (Hsu et al., 2015; Wolfer, 2020). Furthermore, even if mixing time and hitting time are known, by definition of trajectory length, H, the minimum sample budget, T, required for the number of updates, K, to be at least one is practically infeasible as we will explain in Section 4. In this work, we utilize a variant of the actor-critic (AC) algorithm that is able to estimate the advantage with a trajectory length scheme that has no dependence on mixing time, hitting time, and total sample budget as we will explain in more detail in the following sections.

3.3. Actor-Critic Algorithm

The AC algorithm alternates between a actor and a critic. The actor function is the parameterized policy π_θ and the critic function estimates the value function $V^{\pi_\theta}(s)$. We can rewrite the policy gradient theorem in terms of the *temporal difference* (TD), δ^{π_θ}

$$\nabla_\theta J(\theta) = \mathbf{E}_{s\sim d^{\pi_\theta}, a\sim\pi_\theta(\cdot|s), s'\sim p(\cdot|s,a)}\Big[\delta^{\pi_\theta}\nabla_\theta \log \pi_\theta(a|s)\Big], \tag{12}$$

where $\delta^{\pi_\theta} = r(s,a) - J(\theta) + V^{\pi_\theta}(s') - V^{\pi_\theta}(s)$. We can see that the TD is an estimation of the advantage term. Actor-critic algorithms provide a greater stability that alternative PG algoirhtms, such as REINFORCE (Williams, 1992) and PPGAE that estimate the advantage with a sum of observed rewards. The stability comes from the learned value estimator, the critic function, being used as a baseline to reduce variance in the gradient estimation.

Because the scope of this work focuses on the global convergence for the actor, we assume that the critic function is the inner product between a given feature map $\phi(s) : \mathcal{S} \to \mathbb{R}^m$ and a weight vector $\omega \in \mathbb{R}^m$. This assumption allows the critic optimization problem we describe below to be strongly convex.

We denote the critic estimation for $V^{\pi_\theta}(s)$ as $V_\omega(s) := \langle \phi(s), \omega \rangle$ and assume that $\|\phi(s)\| \leq 1$ for all $s \in \mathcal{S}$. The critic aims to minimize the error below,

$$\min_{\omega\in\Omega} \sum_{s\in S} d^{\pi_\theta}(s)(V^{\pi_\theta}(s) - V_\omega(s))^2. \tag{13}$$

By weighting the summation by $d^{\pi_\theta}(s)$, it is more imperative to find an ω that accurately estimates of V value at states where the agent has a higher probability of being in the long-run. The gradient update for ω is given as

$$\begin{aligned}\omega_{t+1} =&\Pi_\Omega\big[\omega_t - \beta_t\big(r(s_t,a_t) - J(\pi_{\theta_t}) + \langle\phi(s_{t+1}),\omega_t\rangle \\ &- \langle\phi(s_t),\omega_t\rangle\big)\phi(s_t)\big],\end{aligned} \tag{14}$$

where β_t is the critic learning rate. Because the critic update in (14) relies on $J(\pi_{\theta_t})$, which we do not have access to, we can substitute a recursive estimate for the average reward as $\eta_{t+1} = \eta_t - \gamma_t(\eta_t - r(s_t,a_t))$. We now write the AC updates as

$$\begin{aligned}\eta_{t+1} =&\eta_t - \gamma_t \cdot f_t, && \text{(reward tracking)}\\ \omega_{t+1} =&\Pi_\Omega\big[\omega_t - \beta_t \cdot g_t\big], && \text{(critic update)}\\ \theta_{t+1} =&\theta_t + \alpha_t \cdot h_t, && \text{(actor update)}\end{aligned} \tag{15}$$

where we have

$$\begin{aligned}f_t =&\eta_t - r(s_t,a_t),\\ g_t =&\big(r(s_t,a_t) - \eta_t + \langle\phi(s_{t+1}) - \phi(s_t),\omega_t\rangle\big)\phi(s_t),\\ h_t =&\delta^{\pi_{\theta_t}} \cdot \nabla_\theta \log \pi_{\theta_t}(a_t|s_t),\\ \delta^{\pi_{\theta_t}} =&r(s_t,a_t) - \eta_t + \langle\phi(s_{t+1}) - \phi(s_t),\omega_t\rangle.\end{aligned} \tag{16}$$

As the critic and reward tracking are vital to average-reward AC, we incorporate the critic and average-reward tracking errors in our global convergence analysis of the actor. One drawback of most vanilla AC algorithms is their assumption on the decay rates of mixing time. Under ergodicity, Markov Chains induced by π_θ reach their respective d^{π_θ} exponentially fast, i.e., for some $\rho \in [0,1], m(t;\theta) \leq \mathcal{O}(\rho^t)$. However, most vanilla AC analyses assume there exists some ρ such that, for all $\theta, m(t;\theta) \leq \mathcal{O}(\rho^t)$. This assumption sets an upper limit on how slow-mixing an environment can be for the algorithm to handle. In the following section, we provide an overview of the AC variant, MAC, that will provide tighter dependence on mixing time despite without oracle knowledge and with no limit on ρ.

3.4. Multi-level Actor-Critic

Recent work has developed the Multi-Level Actor-Critic (MAC) (Suttle et al., 2023) that relies upon a Multi-level Monte-Carlo (MLMC) gradient estimator for the actor, critic, and reward tracking. Let $J_t \sim \text{Geom}(1/2)$ and we collect the trajectory $\mathcal{T}_t := \{s_t^i, a_t^i, r_t^i, s_t^{i+1}\}_{i=1}^{2^{J_t}}$ with policy π_{θ_t}. Then the MLMC policy gradient estimator is given by the following conditional:

$$h_t^{MLMC} = h_t^0 + \begin{cases} 2^{J_t}(h_t^{J_t} - h_t^{J_t-1}), & \text{if } 2^{J_t} \leq T_{\max}\\ 0, & \text{otherwise}\end{cases} \tag{17}$$

with $h_t^j = \frac{1}{2^j}\sum_{i=1}^{2^j} h(\theta_t; s_t^i, a_t^i)$ and where $T_{\max} \geq 2$. We note that the same formula is used for MLMC gradient estimators of the critic, g_t^{MLMC}, and reward tracker, f_t^{MLMC}.

As we will see from Lemma 2, the advantage of the MLMC estimator is that we get the same bias as averaging T gradients with $\widetilde{\mathcal{O}}(1)$ samples. Drawing from a geometric distribution has no dependence on knowing what the mixing time is, thus allowing us to drop the oracle knowledge assumption previously used in works such as (Bai et al., 2024). To reduce the variance introduced by the MLMC estimator, (Dorfman & Levy, 2022; Suttle et al., 2023) used the AdaGrad stepsize scheme (Duchi et al., 2011; Levy, 2017).

4. Global Convergence Analysis

In this section we provide theoretical guarantee of the global convergence of MAC from (Suttle et al., 2023).

4.1. Preliminaries

In this section, we provide assumptions and lemmas that will be used in our global convergence analysis below.

Assumption 1. For all θ, the parameterized MDP $\mathcal{M}_\theta$ induces an ergodic Markov Chain.

Assumption 1 is typical in many works such as (Suttle et al., 2023; Pesquerel & Maillard, 2022; Gong & Wang, 2020; Bai et al., 2024). As previously mentioned, it ensures all states are reachable, and importantly also guarantees the existence of a stationary distribution d^{π_θ} for any induced Markov Chain.

Because we parameterize the critic using linear function approximation, for a fixed policy parameter θ the temporal difference will converge to the minimum of the mean squared projected Bellman error (MSPBE), as discussed in (Sutton & Barto, 2018).

Definition 2. Denoting $\omega^*(\theta)$ as the fixed point for a given θ, and for a given feature mapping ϕ for the critic, we define the worst-case approximation error to be

$$\mathcal{E}_{app}^{critic} = \sup_\theta \sqrt{\mathbf{E}_{s\sim\mu_\theta}\left[\phi(s)^T\omega^*(\theta) - V^{\pi_\theta}(s)\right]^2}, \tag{18}$$

which we assume to be finite. With a well-designed feature map, $\mathcal{E}_{app}^{critic}$ will be small or even 0. We will later see that by assuming $\mathcal{E}_{app}^{critic} = 0$ we recover the $\widetilde{\mathcal{O}}\left(T^{-\frac{1}{4}}\right)$ dependence as in (Bai et al., 2024).

Assumption 2. Let $\{\pi_\theta\}_{\theta\in\mathbb{R}^d}$ denote our parameterized policy class. There exist $B, R, L > 0$ such that

1. $\|\nabla \log \pi_\theta(a|s)\| \leq B$, for all $\theta \in \mathbb{R}^d$,
2. $\|\nabla \log \pi_\theta(a|s) - \nabla \log \pi_{\theta'}(a|s)\| \leq R\|\theta - \theta'\|$, for all $\theta, \theta' \in \mathbb{R}^d$,
3. $|\pi_\theta(a|s) - \pi_{\theta'}(a|s)| \leq L\|\theta - \theta'\|$, for all $\theta, \theta' \in \mathbb{R}^d$.

Assumption 2 establishes regularization conditions for policy gradient ascent and has been widely used in prior work,

including (Suttle et al., 2023; Papini et al., 2018; Kumar et al., 2019; Zhang et al., 2020; Xu et al., 2020; Bai et al., 2024). This assumption is vital for presenting our modified general framework for non-constant stepsize in Lemma 6, as B, R, L will appear in our bound for the difference between optimal reward and the average cumulative reward.

Assumption 3. Define the transferred policy function approximation error as

$$L_{d_\rho^{\pi^*},\pi^*}(h_\theta^*,\theta) = \mathbf{E}\big[\big(\nabla_\theta \log \pi_\theta(a|s)\cdot h_\theta^* - A^{\pi_\theta}(s,a)\big)^2\big], \tag{19}$$

where expectation is over $s\sim d_\rho^{\pi^*}, a\sim\pi^*(\cdot|s)$, π^* is the optimal policy, and h_θ^* is given by

$$h_\theta^* = \arg\min_{h\in\mathbb{R}^d} \mathbf{E}_{s\sim d_\rho^{\pi_\theta}} \mathbf{E}_{a\sim\pi_\theta(\cdot|s)}\Big[\Big(\nabla_\theta \log \pi_\theta(a|s)\cdot h - A^{\pi_\theta}(s,a)\Big)^2\Big]. \tag{20}$$

We assume that the error satisfies $L_{d_\rho^{\pi^*},\pi^*}(h_\theta^*,\theta)\le \mathcal{E}_{app}^{actor}$ for any $\theta\in\Theta$, where $\mathcal{E}_{app}^{actor}$ is a positive constant.

Assumption 3 bounds the error that arises from the policy class parameterization. For neural networks, $\mathcal{E}_{app}^{actor}$ has been shown to be small (Wang et al., 2019), while for softmax policies, $\mathcal{E}_{app}^{actor}=0$ (Agarwal et al., 2021). This approximation assumption has also been used in (Bai et al., 2024) and is important to generalizing the policy parameterization in the convergence analysis, since, as we will later see in Section 4, $\mathcal{E}_{app}^{actor}$ will appear as an independent term in the final bound.

For our later analysis, we also define recall the definition of the Fisher information matrix $F(\theta)$:

$$F(\theta) = \mathbf{E}_{s\sim d^{\pi_\theta}} \mathbf{E}_{a\sim\pi_\theta(\cdot|s)}\left[\nabla_\theta \log\pi_\theta(a|s)(\nabla_\theta\log\pi_\theta(a|s))^T\right].$$

We can now also express h_θ^* defined in (20) as

$$h_\theta^* = F(\theta)^\dagger \mathbf{E}_{s\sim d^{\pi_\theta}} \mathbf{E}_{a\sim\pi_\theta(\cdot|s)}\left[\nabla_\theta \log\pi_\theta(a|s) A^{\pi_\theta}(s,a)\right],$$

where $\dagger$ denotes the Moore-Penrose pseudoinverse.

Assumption 4. Setting I_F as the identity matrix of the same dimension as $F(\theta)$, let there exist some positive constant μ_F such that $F(\theta)-\mu_F I_F$ is positive semidefinite.

Assumption 4 is a common assumption for global convergence of policy gradient algorithms (Liu et al., 2020; Bai et al., 2024). This assumption will be useful later in our analysis by translating the general framework proposed in Lemma 6 into terms of $\|\nabla J(\theta_t)\|^2$, which allows us to leverage the analysis of the local convergence rate of MAC given in (Suttle et al., 2023). We then can use the convergence rate bound established by (Suttle et al., 2023), which is stated in Lemma 4 below.

Lemma 2. *Let $j_{max}=\lfloor \log T_{max}\rfloor$. Fix θ_t measurable w.r.t. $\mathcal{F}_{t-1}$. Assume $T_{max}\ge \tau_{mix}^{\theta_t}$, $\|\nabla J(\theta)\|\le G_H$, for all θ, and $\|h_t^N\|\le G_H$, for all $N\in[T_{max}]$. Then*

$$\mathbf{E}_{t-1}\left[h_t^{MLMC}\right] = \mathbf{E}_{t-1}\left[h_t^{j_{max}}\right], \tag{21}$$

$$\mathbf{E}\left[\|h_t^{MLMC}\|^2\right] \le \widetilde{\mathcal{O}}\left(G_H^2\tau_{mix}^{\theta_t}\log T_{max}\right) + 8\log(T_{max})T_{max}\left(\mathcal{E}(t)+ 16B^2(\mathcal{E}_{app}^{critic})^2\right), \tag{22}$$

$$\mathbf{E}\|\nabla J(\theta_t) - h_t^{j_{max}}\|^2 \le \widetilde{\mathcal{O}}\left(G_H^2\tau_{mix}^{\theta_t}\frac{\log T_{max}}{T_{max}}\right) + \mathcal{E}_2(t) + 16B^2(\mathcal{E}_{app}^{critic})^2. \tag{23}$$

Lemma 2 provides a bound for the variance of the MLMC gradient estimator and will be integral to our analysis.

Lemma 3. *Let $\beta_t = \gamma_t = (1+t)^{-\nu}, \alpha = \alpha_t'/\sqrt{\sum_{k=1}^t \|h_t^{MLMC}\|^2}$, and $\alpha_t' = (1+t)^{-\sigma}$, where $0<\nu<\sigma<1$. Then*

$$\frac{1}{T}\sum_{t=1}^T \mathcal{E}(t) \le \mathcal{O}\left(T^{\nu-1}\right) + \mathcal{O}\left(T^{-2(\sigma-\nu)}\right) + \widetilde{\mathcal{O}}\left(\tau_{mix}\log T_{\max}\right)\mathcal{O}\left(T^{-\nu}\right) + \widetilde{\mathcal{O}}\left(\tau_{mix}\frac{\log T_{\max}}{T_{\max}}\right). \tag{24}$$

By setting $\nu=0.5$ and $\sigma=0.75$ leads to the following:

$$\frac{1}{T}\sum_{t=1}^T \mathcal{E}(t) \le \widetilde{\mathcal{O}}\left(\tau_{mix}\log T_{\max}\right)\mathcal{O}\left(T^{-\frac{1}{2}}\right) + \widetilde{\mathcal{O}}\left(\tau_{mix}\frac{\log T_{\max}}{T_{\max}}\right). \tag{25}$$

Lemma 3, which is established by (Suttle et al., 2023), states the convergence rate of the critic with an MLMC estimator. This result directly affects the following overall MAC convergence rate from (Suttle et al., 2023):

Lemma 4. *(MAC Convergence Rate) Assuming $J(\theta)$ is L-smooth, $\sup_\theta|J(\theta)|\le M$, and $\|\nabla J(\theta)\|, \|h_t^{MLMC}\|\le G_H$, then, for all θ, t and under the assumptions of Lemma 3, we have that*

$$\frac{1}{T}\sum_{t=1}^T \mathbb{E}\left[\|\nabla J(\theta_t)\|^2\right] \le \mathcal{O}\left(\mathcal{E}_{app}^{critic}\right) + \widetilde{\mathcal{O}}\left(\frac{\tau_{mix}\log T_{\max}}{\sqrt{T}}\right) + \widetilde{\mathcal{O}}\left(\frac{\tau_{mix}\log T_{\max}}{T_{\max}}\right). \tag{26}$$

Both Lemmas 3 and 4 rely on the convergence of the error in average reward tracking, provided in Lemma D.1 of (Suttle et al., 2023). However, we have noticed that there is a $\widetilde{\mathcal{O}}\left(\sqrt{\frac{\tau_{mix}\log T_{\max}}{T_{\max}}}\right)$ term in Lemma D.1 that should actually absorb the $\widetilde{\mathcal{O}}\left(\frac{\tau_{mix}\log T_{\max}}{T_{\max}}\right)$ term in (25) and (26).

We provide a correct version of the proof in the Appendix, where we were able to remove the square root.

Finally, we will use the following result to manipulate the AdaGrad stepsizes in the final result of this section.

Lemma 5. *Lemma 4.2, (Dorfman & Levy, 2022). For any non-negative real numbers* $\{a_i\}_{i\in[n]}$,

$$\sum_{i=1}^{n} \frac{a_i}{\sqrt{\sum_{j=1}^{i} a_j}} \leq 2\sqrt{\sum_{i=1}^{n} a_i}. \tag{27}$$

4.2. Global Convergence Guarantee

To develop our convergence analysis, we present a modified version of the general framework proposed in (Bai et al., 2024) to accommodate non-constant stepsizes such as those used in AdaGrad with MLMC gradient estimator h_t^{MLMC}.

Lemma 6. *Suppose an MLMC gradient ascent algorithm updates the policy parameter in the following way, using* $h_t^{MLMC} = h_t$:

$$\theta_{t+1} = \theta_t + \alpha_t h_t. \tag{28}$$

When Assumptions 2, 3, and 4 hold, we have the following inequality for any T:

$$\begin{aligned} J^* - \frac{1}{T}\sum_{t=1}^{T} J(\theta_t) \leq \sqrt{\mathcal{E}_{app}^{actor}} + \frac{B}{T}\sum_{t}^{T} \|(h_t - h_t^*)\| \\ + \frac{R}{2T}\sum_{t=1}^{T} \alpha_t \|h_t\|^2 + \frac{1}{T}\sum_{t=1}^{T} \frac{1}{\alpha_t} \mathbf{E}_{s\sim d^{\pi^*}} \zeta_t, \end{aligned} \tag{29}$$

where $h_t^* := h_{\theta_t}^*$ *and* $h_{\theta_t}^*$ *is defined in (20),* $J^* = J(\theta^*)$, $\pi^* = \pi_{\theta^*}$, *where* θ^* *is the optimal parameter, and* $\zeta_t = [KL(\pi^*(\cdot|s)\|\pi_{\theta_k}(\cdot|s)) - KL(\pi^*(\cdot|s)\|\pi_{\theta_{k+1}}(\cdot|s))]$.

We provide a proof of the above lemma in Appendix A. The proof is similar to that of (Bai et al., 2024) with a notable difference that the non-constant stepsize does not allow us to simplify the telescoping summation in the last term. As will be seen later on in the analysis, we address this issue by bounding α_t with a suitable constant.

Theorem 1. *Let* $\{\theta_t\}_{t=1}^T$ *be defined as in Lemma 6. If assumptions 1, 2, 3, 4 hold and* $J(\cdot)$ *is L-smooth, then the following inequality holds.*

$$\begin{aligned} J^* - \frac{1}{T}\sum_{t=1}^{T} \mathbf{E}\left[J(\theta_t)\right] \leq \sqrt{\mathcal{E}_{app}^{actor}} + \widetilde{\mathcal{O}}\left(\frac{\sqrt{\tau_{mix} T_{\max}} \log T_{\max}}{T^{\frac{1}{2}}}\right) \\ + \widetilde{\mathcal{O}}\left(\frac{\sqrt{\tau_{mix} \log T_{\max}}}{T^{\frac{1}{4}}}\right) + \mathcal{E}_{app}^{critic} \\ + \widetilde{\mathcal{O}}\left(\sqrt{\frac{\tau_{mix} \log T_{\max}}{T_{\max}}}\right). \end{aligned} \tag{30}$$

The proof of the above can be found in Appendix B. We here provide a proof sketch to highlight the main mechanics.

Proof sketch. We first rewrite the bound on the expectation in (29) into terms of (22) of Lemma 2 and (26) of Lemma 4. The expectation in the second term on the right-hand side (RHS) of (29) can then be bounded as follows using Lemma 2 and Assumption 4:

$$\begin{aligned} \frac{1}{T}\sum_{t=1}^{T} \mathbf{E}\|h_t - h_t^*\| \leq \sqrt{\frac{1}{T}\sum_{t=1}^{T} \mathbf{E}\|h_t^{j_{max}} - \nabla J(\theta_t)\|^2} \\ + \sqrt{\frac{2}{T}\sum_{t=1}^{T}\left(2 + \frac{1}{\mu_F^2}\right)\mathbf{E}\left[\|\nabla_\theta J(\theta_t)\|^2\right]}. \end{aligned} \tag{31}$$

For the third term of the RHS of (29), we can use Lemma 5 to obtain that

$$\frac{R}{2T}\sum_{t=1}^{T} \alpha_t \|h_t\|^2 \leq \frac{R}{T}\sqrt{\sum_{t=1}^{T} \|h_t\|^2}. \tag{32}$$

We bound the fourth term as follows, using the fact that it is a telescoping sum and that $\alpha_T < \alpha_t$, $\alpha_T = \frac{\alpha_T'}{\sum_{t=1}^T \|h_t\|^2}$:

$$\begin{aligned} \frac{1}{T}\sum_{t=1}^{T} \frac{1}{\alpha_t}\mathbf{E}_{s\sim d^{\pi^*}}[\zeta_t] \leq \frac{\mathbf{E}_{s\sim d^{\pi^*}}[KL(\pi^*(\cdot|s)\|\pi_{\theta_1}(\cdot|s))]}{\alpha_T'} \\ \times \frac{1}{T}\sqrt{\sum_{t=1}^{T} \|h_t\|^2}. \end{aligned} \tag{33}$$

Plugging these bounds back into (29) and ignoring constants, we obtain that

$$\begin{aligned} J^* - \frac{1}{T}\sum_{t=1}^{T} \mathbf{E}\|J(\theta_t)\| \leq \sqrt{\mathcal{E}_{app}^{actor}} + \frac{1}{T}\sqrt{\sum_{t=1}^{T} \mathbf{E}\left[\|h_t\|^2\right]} \\ + \sqrt{\frac{1}{T}\sum_{t=1}^{T} \mathbf{E}\|h_t^{j_{max}} - \nabla J(\theta_t)\|^2} \\ + \sqrt{\frac{1}{T}\sum_{t=1}^{T} \mathbf{E}\left[\|\nabla_\theta J(\theta_t)\|^2\right]}. \end{aligned} \tag{34}$$

Bounding the the second and third term by the RHS with Lemmas 2 and 4 respectively concludes the proof.

Remark. With Theorem 1, we can recover the $\mathcal{O}\left(T^{-\frac{1}{4}}\right)$ as in (Bai et al., 2024). Furthermore, MAC has a tighter dependence on mixing time with $\widetilde{\mathcal{O}}\left(\sqrt{\tau_{mix}}\right)$ compared to the $\widetilde{\mathcal{O}}\left(\tau_{mix}^2\right)$ in (Bai et al., 2024), despite having no prior knowledge of mixing time due to the combination of MLMC gradient estimation and AdaGrad stepsize. Similar to (Bai et al., 2024), the independent term $\mathcal{E}_{app}^{actor} \geq 0$ accounts for the general policy parameterization. However, our bound has no dependence on hitting time like that in (Bai et al., 2024), as their dependence was a result of their advantage estimation algorithm described in Section 3.2.

Discussion on Practicality. In this section, we want to highlight how practically feasible it is to implement MAC as compared to PPGAE. As mentioned previously, PPGAE defines T as the sample budget of the entire training process, K as the number of gradient updates, and $H = 16\tau_{hit}\tau_{mix}\sqrt{T}(\log(T))^2$ as the length of one update, whence $K = T/H$. Since K represents the number of updates, it must be a positive integer. For $K = T/H \leq 1$, this is equivalent to $\frac{\sqrt{T}}{(\log(T))^2} \leq 16\tau_{hit}\tau_{mix}$. Even if $\tau_{hit} = 10$ and $\tau_{mix} = 1$, $H \approx 6.6 * 10^9$. Since τ_{hit} can become infinitely large and τ_{mix} grows with environment complexity, in practice H would be much higher. As Figure 1 shows, if we set $\tau_{hit} = 10$, then, as mixing time increases, the minimum episode length H increases exponentially to satisfy $K > 1$. Even at $\tau_{mix} = 60$, the minimum H is around 10^{14} samples.

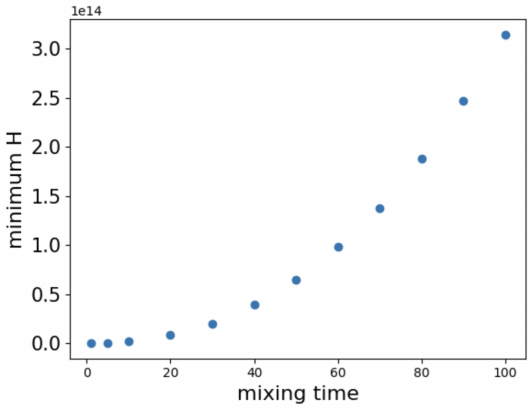

Figure 1. Minimum H required for $K = 1$ given a mixing time τ_{mix}. Both H and τ_{mix} are in terms of number of samples. We set the hitting time to be 10 for this plot.

In contrast, for MAC the trajectory length is based on a geometric distribution with no dependence on mixing time, hitting time, or total sample budget.

4.3. Experimental Results

As a preliminary proof-of-concept experiment to show the advantage of MAC over PPGAE, Vanilla AC, and REINFORCE, we consider a 15-by-15 sparse gridworld environment. The agent tries to from the top left to bottom right corner. The agent receives a reward of $+1$ if goal is reached and $+0$ else. The episode ends when the agent either reaches the goal or hits a limit of 200 samples. We report a moving average success rate over 100 trials with 95% confidence intervals in Figure 2. We can see that MAC has a higher success rate and can more consistently reach the goal than the baselines.

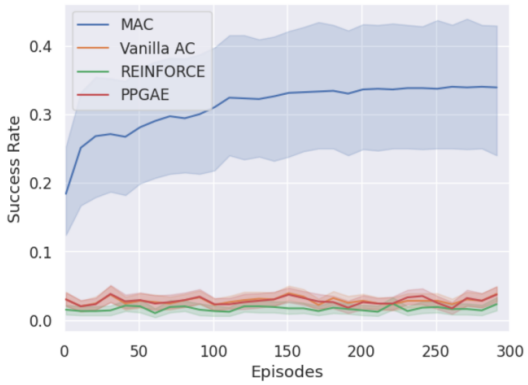

Figure 2. Success Rate in a sparse 15-by-15 grid over 300 training episodes with 200 samples per episode. For MAC, $T_{max} = 4$ and for PPGAE, $H = 200$ and $N = 1$. Vanilla AC and REINFORCE both have $H = 200$. and 100 trials for each algorithm. PPGAE, Vanilla AC, and REINFORCE consistently converge to significantly less optimal solutions than MAC.

5. Conclusion and Further Work

In this work, we provide policy gradient global convergence analysis for the infinite horizon average reward MDP without restrictive and impractical assumptions on mixing time. Using MAC, we show that actor-critic models, utilizing a MLMC gradient estimator, achieves a tighter dependence on mixing time for global convergence. We hope this work encourages further investigation into algorithms that do not assume oracle knowledge of mixing time. Future work can also further test the advantages of MAC in slow mixing environments for robotics, finance, healthcare, and other applications.

Followed by this work, (Ganesh et al., 2024) has proposed an algorithm that achieves convergence rate of $\widetilde{\mathcal{O}}\left(T^{-\frac{1}{2}}\right)$, while requiring knowledge of mixing time. Coming up with such guarantees without knowledge of mixing time is an open problem.

Acknowledgements

This research was supported by Army Cooperative Agreement W911NF2120076.

Disclaimer

This paper was prepared for informational purposes in part by the Artificial Intelligence Research group of JP Morgan Chase & Co and its affiliates ("JP Morgan"), and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document

is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

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, 2020.

Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *The Journal of Machine Learning Research*, 22(1):4431–4506, 2021.

Bai, Q., Mondal, W. U., and Aggarwal, V. Regret analysis of policy gradient algorithm for infinite horizon average reward markov decision processes. In *AAAI Conference on Artificial Intelligence*, 2024.

Bedi, A. S., Chakraborty, S., Parayil, A., Sadler, B. M., Tokekar, P., and Koppel, A. On the hidden biases of policy mirror ascent in continuous action spaces. In *International Conference on Machine Learning*, pp. 1716–1731, 2022.

Bhandari, J. and Russo, D. Global optimality guarantees for policy gradient methods. *Operations Research*, 2024.

Dorfman, R. and Levy, K. Y. Adapting to mixing time in stochastic optimization with Markovian data. In *Proceedings of the 39th International Conference on Machine Learning*, Jul 2022.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12 (61):2121–2159, 2011. URL http://jmlr.org/papers/v12/duchi11a.html.

Duchi, J. C., Agarwal, A., Johansson, M., and Jordan, M. I. Ergodic mirror descent. *SIAM Journal on Optimization*, 22(4):1549–1578, 2012.

Ganesh, S., Mondal, W. U., and Aggarwal, V. Variance-reduced policy gradient approaches for infinite horizon average reward markov decision processes. *arXiv preprint arXiv:2404.02108*, 2024.

Gaur, M., Aggarwal, V., and Agarwal, M. On the global convergence of fitted q-iteration with two-layer neural network parametrization. In *International Conference on Machine Learning*, pp. 11013–11049. PMLR, 2023.

Gaur, M., Aggarwal, V., Bedi, A. S., and Wang, D. Closing the gap: Achieving global convergence (last iterate) of actor-critic under markovian sampling with neural network parametrization. In *ICML*, 2024.

Geng, N., Lan, T., Aggarwal, V., Yang, Y., and Xu, M. A multi-agent reinforcement learning perspective on distributed traffic engineering. In *2020 IEEE 28th International Conference on Network Protocols (ICNP)*, pp. 1–11. IEEE, 2020.

Gong, H. and Wang, M. A duality approach for regret minimization in average-award ergodic markov decision processes. In *Learning for Dynamics and Control*, 2020.

Hsu, D. J., Kontorovich, A., and Szepesvári, C. Mixing time estimation in reversible markov chains from a single sample path. *Advances in neural information processing systems*, 28, 2015.

Kumar, H., Koppel, A., and Ribeiro, A. On the sample complexity of actor-critic method for reinforcement learning with function approximation. *arXiv preprint arXiv:1910.08412*, 2019.

Levy, K. Online to offline conversions, universality and adaptive minibatch sizes. *Advances in Neural Information Processing Systems*, 30, 2017.

Ling, L., Mondal, W. U., and Ukkusuri, S. V. Cooperating graph neural networks with deep reinforcement learning for vaccine prioritization. *arXiv preprint arXiv:2305.05163*, 2023.

Liu, Y., Zhang, K., Basar, T., and Yin, W. An improved analysis of (variance-reduced) policy gradient and natural policy gradient methods. *Advances in Neural Information Processing Systems*, 33:7624–7636, 2020.

Mei, J., Xiao, C., Szepesvari, C., and Schuurmans, D. On the global convergence rates of softmax policy gradient methods. In *International Conference on Machine Learning*, pp. 6820–6829, 2020.

Mondal, W. U. and Aggarwal, V. Improved sample complexity analysis of natural policy gradient algorithm with general parameterization for infinite horizon discounted

reward markov decision processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 3097–3105. PMLR, 2024.

Nagaraj, D., Wu, X., Bresler, G., Jain, P., and Netrapalli, P. Least squares regression with markovian data: Fundamental limits and algorithms. In *Advances in Neural Information Processing Systems*, 2020.

Papini, M., Binaghi, D., Canonaco, G., Pirotta, M., and Restelli, M. Stochastic variance-reduced policy gradient. In *International conference on machine learning*, pp. 4026–4035, 2018.

Patel, B., Weerakoon, K., Suttle, W. A., Koppel, A., Sadler, B. M., Bedi, A. S., and Manocha, D. Ada-nav: Adaptive trajectory-based sample efficient policy learning for robotic navigation. *arXiv preprint arXiv:2306.06192*, 2023.

Pesquerel, F. and Maillard, O.-A. Imed-rl: Regret optimal learning of ergodic markov decision processes. In *NeurIPS 2022-Thirty-sixth Conference on Neural Information Processing Systems*, 2022.

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

Riemer, M., Raparthy, S. C., Cases, I., Subbaraj, G., Touzel, M. P., and Rish, I. Continual learning in environments with polynomial mixing times. *arXiv preprint arXiv:2112.07066*, 2021.

Suttle, W. A., Bedi, A., Patel, B., Sadler, B. M., Koppel, A., and Manocha, D. Beyond exponentially fast mixing in average-reward reinforcement learning via multi-level monte carlo actor-critic. In *International Conference on Machine Learning*, pp. 33240–33267, 2023.

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

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Wang, L., Cai, Q., Yang, Z., and Wang, Z. Neural policy gradient methods: Global optimality and rates of convergence. In *International Conference on Learning Representations*, 2019.

Wei, C.-Y., Jahromi, M. J., Luo, H., and Jain, R. Learning infinite-horizon average-reward mdps with linear function approximation. In *International Conference on Artificial Intelligence and Statistics*, pp. 3007–3015. PMLR, 2021.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Wolfer, G. Mixing time estimation in ergodic markov chains from a single trajectory with contraction methods. In *Proceedings of the 31st International Conference on Algorithmic Learning Theory*, Feb 2020.

Xu, P., Gao, F., and Gu, Q. An improved convergence analysis of stochastic variance-reduced policy gradient. In *Uncertainty in Artificial Intelligence*, pp. 541–551, 2020.

Zhang, K., Koppel, A., Zhu, H., and Basar, T. Global convergence of policy gradient methods to (almost) locally optimal policies. *SIAM Journal on Control and Optimization*, 58(6):3586–3612, 2020.

Zhang, Y. and Ross, K. W. On-policy deep reinforcement learning for the average-reward criterion. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12535–12545. PMLR, 2021.

Contents

Appendix

A. Proof of Lemma 6

In this section, we provide a bound for the difference between the optimal reward and the cumulative reward observed up to trajectory T that will be used as our general framework for the global convergence analysis. Our framework is a modification from (Bai et al., 2024) in that it can handle non-constant stepsizes. The framework provided in (Bai et al., 2024) is itself an average reward adaptation of the framework provided by (Liu et al., 2020) for the discounted reward setting. We first provide a supporting result in the form the average reward performance difference lemma:

Lemma 7. *The difference in the performance for any policies π_θ and $\pi_{\theta'}$ is bounded as follows*

$$J(\theta) - J(\theta') = \mathbb{E}_{s\sim d^{\pi_\theta}} \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\big[A^{\pi_{\theta'}}(s,a)\big], \tag{35}$$

We can now provide the general framework lemma.

Lemma 8. *Suppose a general gradient ascent algorithm updates the policy parameter in the following way.*

$$\theta_{t+1} = \theta_t + \alpha_t h_t. \tag{36}$$

When Assumptions 2, 3, and 7 hold, we have the following inequality for any T.

$$J^* - \frac{1}{T}\sum_{t=1}^{T} J(\theta_t) \leq \sqrt{\mathcal{E}_{app}^{actor}} + \frac{B}{T}\sum_{t}^{T} \|(h_t - h_t^*)\| + \frac{K}{2T}\sum_{t=1}^{T}\alpha_t\|h_t\|^2 + \frac{1}{T}\sum_{t=1}^{T}\frac{1}{\alpha_t}\mathbb{E}_{s\sim d^{\pi^*}}\zeta_t, \tag{37}$$

where $h_t^* := h_{\theta_t}^*$ and $h_{\theta_t}^*$ is defined in (20), $J^* = J(\theta^*)$, and $\pi^* = \pi_{\theta^*}$ where θ^* is the optimal parameter, and $\zeta_t = [KL(\pi^*(\cdot|s)\|\pi_{\theta_t}(\cdot|s)) - KL(\pi^*(\cdot|s)\|\pi_{\theta_{t+1}}(\cdot|s))]$.

Proof. We start the proof by lower bounding the difference between the KL divergence between $\pi*$ and π_θ and the KL divergence between $\pi*$ and $\pi_{\theta+1}$.

$$\mathbb{E}_{s\sim d^{\pi^*}}[KL(\pi^*(\cdot|s)\|\pi_{\theta_t}(\cdot|s)) - KL(\pi^*(\cdot|s)\|\pi_{\theta_{t+1}}(\cdot|s))] \tag{38}$$

$$= \mathbb{E}_{s\sim d^{\pi^*}}\mathbb{E}_{a\sim\pi^*(\cdot|s)}\left[\log\frac{\pi_{\theta_{t+1}}(a|s)}{\pi_{\theta_t}(a|s)}\right] \tag{39}$$

$$\overset{(a)}{\geq} \mathbb{E}_{s\sim d^{\pi^*}}\mathbb{E}_{a\sim\pi^*(\cdot|s)}[\nabla_\theta \log\pi_{\theta_t}(a|s)\cdot(\theta_{t+1}-\theta_t)] - \frac{K}{2}\|\theta_{t+1}-\theta_t\|^2 \tag{40}$$

$$= \mathbb{E}_{s\sim d^{\pi^*}}\mathbb{E}_{a\sim\pi^*(\cdot|s)}[\nabla_\theta \log\pi_{\theta_t}(a|s)\cdot\alpha_t h_t] - \frac{K\alpha_t^2}{2}\|h_t\|^2 \tag{41}$$

$$= \mathbb{E}_{s\sim d^{\pi^*}}\mathbb{E}_{a\sim\pi^*(\cdot|s)}[\nabla_\theta \log\pi_{\theta_t}(a|s)\cdot\alpha_t h_t^*] + \mathbb{E}_{s\sim d^{\pi^*}}\mathbb{E}_{a\sim\pi^*(\cdot|s)}[\nabla_\theta \log\pi_{\theta_t}(a|s)\cdot\alpha_t(h_t-h_t^*)] - \frac{K\alpha_t^2}{2}\|h_t\|^2 \tag{42}$$

$$= \alpha_t[J^* - J(\theta_t)] + \mathbb{E}_{s\sim d^{\pi^*}}\mathbb{E}_{a\sim\pi^*(\cdot|s)}[\nabla_\theta \log\pi_{\theta_t}(a|s)\cdot\alpha_t h_t^*] - \alpha_t[J^* - J(\theta_t)] \tag{43}$$

$$+ \mathbb{E}_{s\sim d^{\pi^*}}\mathbb{E}_{a\sim\pi^*(\cdot|s)}[\nabla_\theta \log\pi_{\theta_t}(a|s)\cdot\alpha_t(h_t-h_t^*)] - \frac{K\alpha_t^2}{2}\|h_t\|^2 \tag{44}$$

$$\overset{(b)}{=} \alpha_t[J^* - J(\theta_t)] + \alpha_t\mathbb{E}_{s\sim d^{\pi^*}}\mathbb{E}_{a\sim\pi^*(\cdot|s)}\left[\nabla_\theta \log\pi_{\theta_t}(a|s)\cdot h_t^* - A^{\pi_{\theta_t}}(s,a)\right] \tag{45}$$

$$+ \alpha_t\mathbb{E}_{s\sim d^{\pi^*}}\mathbb{E}_{a\sim\pi^*(\cdot|s)}[\nabla_\theta \log\pi_{\theta_t}(a|s)\cdot(h_t-h_t^*)] - \frac{K\alpha_t^2}{2}\|h_t\|^2. \tag{46}$$

Taking the conditional expectation in the above expression, and using the equality in (21), we can write:

$$\mathbb{E}_t\mathbb{E}_{s\sim d^{\pi^*}}[KL(\pi^*(\cdot|s)\|\pi_{\theta_t}(\cdot|s)) - KL(\pi^*(\cdot|s)\|\pi_{\theta_{t+1}}(\cdot|s))] \tag{47}$$

$$\geq \alpha_t[J^* - \mathbb{E}_t[J(\theta_t)]] + \alpha_t\mathbb{E}_t\mathbb{E}_{s\sim d^{\pi^*}}\mathbb{E}_{a\sim\pi^*(\cdot|s)}\left[\nabla_\theta \log\pi_{\theta_t}(a|s)\cdot h_t^* - A^{\pi_{\theta_t}}(s,a)\right] \tag{48}$$

$$+ \alpha_t\mathbb{E}_{s\sim d^{\pi^*}}\mathbb{E}_{a\sim\pi^*(\cdot|s)}[\nabla_\theta \log\pi_{\theta_t}(a|s)\cdot(h_t^{j_{\max}} - h_t^*)] - \frac{K\alpha_t^2}{2}\mathbb{E}_t\|h_t\|^2 \tag{49}$$

$$\overset{(c)}{\geq} \alpha_t[J^* - J(\theta_t)] - \alpha_t\sqrt{\mathbb{E}_{s\sim d^{\pi^*}}\mathbb{E}_{a\sim\pi^*(\cdot|s)}\left[\left(\nabla_\theta \log \pi_{\theta_t}(a|s)\cdot h_t^* - A^{\pi_{\theta_t}}(s,a)\right)^2\right]} \tag{50}$$

$$- \alpha_t\mathbb{E}_{s\sim d^{\pi^*}}\mathbb{E}_{a\sim\pi^*(\cdot|s)}\|\nabla_\theta \log \pi_{\theta_t}(a|s)\|_2\|(h_t^{j_{\max}} - h_t^*)\| - \frac{K\alpha_t^2}{2}\|h_t\|^2 \tag{51}$$

$$\overset{(d)}{\geq} \alpha_t[J^* - J(\theta_t)] - \alpha_t\sqrt{\mathcal{E}_{app}^{actor}} - \alpha_t B\|(h_t^{j_{\max}} - h_t^*)\| - \frac{K\alpha_t^2}{2}\|h_t\|^2, \tag{52}$$

$$\tag{53}$$

where we use Assumption 2 for step (a) and Lemma 7 for step (b). Step (c) uses the convexity of the function $f(x) = x^2$, and (d) comes from Assumption 3. We can get by rearranging terms,

$$\begin{aligned} J^* - J(\theta_t) \leq & \sqrt{\mathcal{E}_{app}^{actor}} + B\|(h_t^{j_{\max}} - h_t^*)\| + \frac{K\alpha_t}{2}\|h_t\|^2 \\ & + \frac{1}{\alpha_t}\mathbb{E}_{s\sim d^{\pi^*}}[KL(\pi^*(\cdot|s)\|\pi_{\theta_t}(\cdot|s)) - KL(\pi^*(\cdot|s)\|\pi_{\theta_{t+1}}(\cdot|s))]. \end{aligned} \tag{54}$$

Because KL divergence is either 0 or positive, we can conclude the proof by taking the average over T trajectories. □

B. Proof of Theorem 1

To use 29 for our convergence analysis, we will take the expectation of the second term. With $h_t^{MLMC} = h_t$, note that,

$$\begin{aligned} \left(\frac{1}{T}\sum_{t=1}^T \mathbb{E}\|h_t^{j_{\max}} - h_t^*\|\right)^2 \leq & \frac{1}{T}\sum_{t=1}^T \mathbb{E}\left[\|h_t^{j_{\max}} - h_t^*\|^2\right] \\ = & \frac{1}{T}\sum_{t=1}^T \mathbb{E}\left[\|h_t^{j_{\max}} - F(\theta_t)^\dagger\nabla_\theta J(\theta_t)\|^2\right] \\ \leq & \frac{2}{T}\sum_{t=1}^T \mathbb{E}\left[\|h_t^{j_{\max}} - \nabla_\theta J(\theta_t)\|^2\right] + \frac{2}{T}\sum_{t=1}^T \mathbb{E}\left[\|\nabla_\theta J(\theta_t) - F(\theta_t)^\dagger\nabla_\theta J(\theta_t)\|^2\right] \\ \overset{(a)}{\leq} & \frac{2}{T}\sum_{t=1}^T \mathbb{E}\left[\|h_t^{j_{\max}} - \nabla_\theta J(\theta_t)\|^2\right] + \frac{2}{T}\sum_{t=1}^T \left(1 + \frac{1}{\mu_F^2}\right)\mathbb{E}\left[\|\nabla_\theta J(\theta_t)\|^2\right], \end{aligned}$$

where (a) uses Assumption 4. Taking the square root of both sides and from $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ we arrive at:

$$\left(\frac{1}{T}\sum_{t=1}^T \mathbb{E}\|h_t^{j_{\max}} - h_t^*\|\right) \leq \sqrt{\frac{2}{T}\sum_{t=1}^T \mathbb{E}\left[\|h_t^{j_{\max}} - \nabla_\theta J(\theta_t)\|^2\right]} + \sqrt{\frac{2}{T}\sum_{t=1}^T \left(1 + \frac{1}{\mu_F^2}\right)\mathbb{E}\left[\|\nabla_\theta J(\theta_t)\|^2\right]}. \tag{55}$$

We can also bound the third term of the RHS of 29 with Lemma 5

$$\frac{R}{2T}\sum_{t=1}^T \alpha_t\|h_t\|^2 \leq \frac{R}{2T}\sum_{t=1}^T \frac{\|h_t\|^2}{\sqrt{\sum_{o=1}^t \|h_o\|^2}} \leq \frac{R}{T}\sqrt{\sum_{t=1}^T \|h_t\|^2}. \tag{56}$$

We can also bound the fourth term using the fact that it is a telescoping sum and that $\alpha_T < \alpha_t$,

$$\begin{aligned} \frac{1}{T}\sum_{t=1}^T \frac{1}{\alpha_t}\mathbb{E}_{s\sim d^{\pi^*}}[\zeta_t] \leq & \frac{1}{T}\sum_{t=1}^T \frac{\mathbb{E}_{s\sim d^{\pi^*}}[\zeta_t]}{\alpha_T} \\ = & \frac{1}{T}\frac{\mathbb{E}_{s\sim d^{\pi^*}}[KL(\pi^*(\cdot|s)\|\pi_{\theta_1}(\cdot|s))]}{\alpha_T} \\ \leq & \frac{\mathbb{E}_{s\sim d^{\pi^*}}[KL(\pi^*(\cdot|s)\|\pi_{\theta_1}(\cdot|s))]\sqrt{\sum_{t=1}^T \|h_t\|^2}}{T\alpha_T'}. \end{aligned} \tag{57}$$

Taking the expectation of both sides of 29 and plugging in 55, 56, and 57, ignoring constants:

$$\begin{aligned} J^* - \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|J(\theta_t)\| \leq & \sqrt{\mathcal{E}_{app}^{actor}} + \sqrt{\frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\|h_t^{j_{max}} - \nabla J(\theta_t)\|^2} \\ & + \sqrt{\frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|\nabla_\theta J(\theta_t)\|^2\right]} + \frac{1}{T}\sqrt{\sum_{t=1}^{T}\mathbf{E}\left[\|h_t\|^2\right]}. \end{aligned} \tag{58}$$

From Lemma 2 we can bound the summation of the expected variance of the MLMC gradient. Ignoring the G_H constant,

$$\sum_{t=1}^{T}\mathbb{E}\left[\|h_t\|^2\right] \leq \sum_{t=1}^{T}\widetilde{\mathcal{O}}\left(\tau_{mix}^{\theta_t}\log T_{max}\right) + \sum_{t=1}^{T}\log(T_{max})T_{max}\mathcal{E}_2(t) + \sum_{t=1}^{T}\log(T_{max})T_{max}(\mathcal{E}_{app}^{critic})^2. \tag{59}$$

We can bound the third term of the RHS by utilizing the maximum mixing time, τ_{mix}

$$\sum_{t=1}^{T}\widetilde{\mathcal{O}}\left(\tau_{mix}^{\theta_t}\log T_{max}\right) \leq \widetilde{\mathcal{O}}\left(T\tau_{mix}\log T_{max}\right). \tag{60}$$

For the second term by using 25,

$$\begin{aligned} \sum_{t=1}^{T}\log(T_{max})T_{max}\mathcal{E}(t) \leq & T(\log T_{\max})T_{\max}\widetilde{\mathcal{O}}\left(\tau_{mix}(\log T_{\max})\right)\mathcal{O}\left(T^{-\frac{1}{2}}\right) \\ & + T(\log T_{\max})T_{\max}\widetilde{\mathcal{O}}\left(\tau_{mix}\frac{(\log T_{\max})}{T_{\max}}\right) \\ = & \widetilde{\mathcal{O}}\left(T\tau_{mix}(\log T_{\max})^2 T_{\max}\right). \end{aligned} \tag{61}$$

The third term can be simply bounded as follows:

$$\sum_{t=1}^{T}\log(T_{max})T_{max}(\mathcal{E}_{app}^{critic})^2 \leq \mathcal{O}\left(T\log(T_{max})T_{max}(\mathcal{E}_{app}^{critic})^2\right). \tag{62}$$

We can now bound the summation of the expected variance of the MLMC gradient:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|h_t\|^2\right] \leq \widetilde{\mathcal{O}}\left(T\tau_{mix}\log T_{max}\right) + \widetilde{\mathcal{O}}\left(T\tau_{mix}(\log T_{\max})^2 T_{\max}\right) + \mathcal{O}\left(T\log(T_{max})T_{max}(\mathcal{E}_{app}^{critic})^2\right). \tag{63}$$

Taking the square root of both sides, from $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$, and dividing by T:

$$\frac{1}{T}\sqrt{\sum_{t=1}^{T}\mathbb{E}\left[\|h_t\|^2\right]} \leq \widetilde{\mathcal{O}}\left(\frac{\sqrt{\tau_{mix}\log T_{max}}}{T^{\frac{1}{2}}}\right) + \widetilde{\mathcal{O}}\left(\frac{\sqrt{\tau_{mix}T_{\max}}\log T_{\max}}{T^{\frac{1}{2}}}\right) + \mathcal{O}\left(\frac{\sqrt{\log(T_{max})T_{max}}\mathcal{E}_{app}^{critic}}{T^{\frac{1}{2}}}\right). \tag{64}$$

From Lemma 4 we can bound the square root of the summation of the expectation of the gradient norm squared:

$$\sqrt{\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla J(\theta_t)\|^2\right]} \leq \mathcal{O}\left(\sqrt{\mathcal{E}_{app}^{critic}}\right) + \widetilde{\mathcal{O}}\left(\frac{\sqrt{\tau_{mix}\log T_{\max}}}{T^{\frac{1}{4}}}\right) + \widetilde{\mathcal{O}}\left(\frac{\sqrt{\tau_{mix}\log T_{\max}}}{\sqrt{T_{\max}}}\right). \tag{65}$$

We can also bound the error between the first term with Equation 23 and Lemma 2:

$$\sqrt{\frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\|h_t^{j_{max}} - \nabla J(\theta_t)\|^2} \leq \mathcal{O}\left(\mathcal{E}_{app}^{critic}\right) + \widetilde{\mathcal{O}}\left(\frac{\sqrt{\tau_{mix}\log T_{\max}}}{T^{\frac{1}{4}}}\right) + \widetilde{\mathcal{O}}\left(\frac{\sqrt{\tau_{mix}\log T_{\max}}}{\sqrt{T_{\max}}}\right). \tag{66}$$

We can see that all terms of (66) absorb the terms of (65). Combining (64) and (66) we can get the final global convergence.

C. Corrected Analysis of Multi-level Monte Carlo

In this section, we wish to provide a corrected analysis for Lemma 4. The issue lies in the convergence rate of the average reward tracker. We first give an overview of the problem and how it affects Lemma 4. We then provide a corrected version of the average reward tracking analysis.

C.1. Overview of Correction

We repeat Lemma 4,

Lemma 9. *Assume $J(\theta)$ is L-smooth, $\sup_\theta |J(\theta)| \leq M$, and $\|\nabla J(\theta)\|, \|h_t^{MLMC}\| \leq G_H$, for all θ, t and under assumptions of Lemma 3, we have*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla J(\theta_t)\|^2\right] \leq \mathcal{O}\left(\mathcal{E}_{app}^{critic}\right) + \widetilde{\mathcal{O}}\left(\frac{\tau_{mix}\log T_{\max}}{\sqrt{T}}\right) + \widetilde{\mathcal{O}}\left(\frac{\tau_{mix}\log T_{\max}}{T_{\max}}\right). \tag{67}$$

The above lemma is correct. However, the analysis for it does not match this final statement. Specifically, given the current analysis provided in (Suttle et al., 2023), the $\widetilde{\mathcal{O}}\left(\frac{\tau_{mix}\log T_{\max}}{T_{\max}}\right)$ term should actually be $\widetilde{\mathcal{O}}\left(\sqrt{\frac{\tau_{mix}\log T_{\max}}{T_{\max}}}\right)$. The term stems from Lemma 3, the convergence of the critic estimation $\mathcal{E}(t)$, which we repeat here,

Lemma 10. *Let $\beta_t = \gamma_t = (1+t)^{-\nu}, \alpha = \alpha_t'/\sqrt{\sum_{k=1}^{t}\|h_t^{MLMC}\|^2}$, and $\alpha_t' = (1+t)^{-\sigma}$, where $0 < \nu < \sigma < 1$. Then*

$$\frac{1}{T}\sum_{t=1}^{T}\mathcal{E}(t) \leq \mathcal{O}\left(T^{\nu-1}\right) + \mathcal{O}\left(T^{-2(\sigma-\nu)}\right) + \widetilde{\mathcal{O}}\left(\tau_{mix}\log T_{\max}\right)\mathcal{O}\left(T^{-\nu}\right) + \widetilde{\mathcal{O}}\left(\tau_{mix}\frac{\log T_{\max}}{T_{\max}}\right). \tag{68}$$

Once again the $\widetilde{\mathcal{O}}\left(\frac{\tau_{mix}\log T_{\max}}{T_{\max}}\right)$ term should actually be $\widetilde{\mathcal{O}}\left(\sqrt{\frac{\tau_{mix}\log T_{\max}}{T_{\max}}}\right)$ based on the current analysis. This term from Lemma 3 is dependent on the average reward tracking error. We repeat its convergence theorem from (Suttle et al., 2023) below,

Theorem 2. *Let $\beta_t = \gamma_t = (1+t)^{-\nu}, \alpha = \alpha_t'/\sqrt{\sum_{k=1}^{t}\|h_t\|^2}$, and $\alpha_t' = (1+t)^{-\sigma}$, where $0 < \nu < \sigma < 1$. Then*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\left[(\eta_t - \eta_t^*)^2\right] \leq \mathcal{O}\left(T^{\nu-1}\right) + \mathcal{O}\left(T^{-2(\sigma-\nu)}\right) + \widetilde{\mathcal{O}}\left(\tau_{mix}\log T_{max}\right)\mathcal{O}\left(T^{-\nu}\right) + \widetilde{\mathcal{O}}\left(\sqrt{\tau_{mix}\frac{\log T_{max}}{T_{max}}}\right). \tag{69}$$

The proof of Theorem 2 matches the statement above. In the next subsection we provide a correct version of the statement along with a proof that will align with Lemmas 3 and 4.

C.2. Corrected Average Reward Tracking Error Analysis

Before we provide the correct version of Theorem 2, we provide the following lemma from (Dorfman & Levy, 2022) and utilized by (Suttle et al., 2023) for Theorem 2 as we will still use it for the correct version of the theorem. In (Suttle et al., 2023), the following lemma is written for MLMC gradient estimator in general. Our restatement is tailored to the MLMC gradient estimation of the reward tracking error.

Lemma 11. *Lemma A.6, (Dorfman & Levy, 2022). Given a policy π_θ, assume we the trajectory sampled from it is $z_t = \{z_t^i = (s_t^i, a_t^i, r_t^i, s_t^{i+1})\}_{i\in[N]}$ starting from $s_t^0 \sim \mu_0(\cdot)$, where μ_0 is the initial state distribution. Let $\nabla F(\eta)$ be an average reward tracking gradient that we wish to estimate over z_t, where $\mathbb{E}_{z\sim\mu_{\theta_t},\pi_{\theta_t}}[f(\eta, z)] = \nabla F(x)$, and $\eta \in \mathcal{K} \subset \mathbb{R}^k$ is the parameter of the estimator. Finally, assume that $\|f(\eta, z)\|, \|\nabla F(\eta)\| \leq 1$, for all $\eta \in \mathcal{K}, z \in \mathcal{S}\times\mathcal{A}\times\mathbb{R}\times\mathcal{S}$. Define $f_t^N = \frac{1}{N}\sum_{i=1}^{N} f(\eta_t, z_t^i)$. Fix $T_{max} \in \mathbb{N}$ and let $K = \tau_{mix}\lceil 2\log T_{max}\rceil$. Then, for every $N \in [T_{max}]$ and every $\eta_t \in \mathcal{K}$ measurable w.r.t. $\mathcal{F}_{t-1} = \sigma(\theta_k, \eta_k, \omega_k, z_k; k \leq t-1)$, where θ and ω are the parameters of the actor and critic, respectively,*

$$\mathbb{E}\left[\|f_t^N - \nabla F(\eta_t)\|\right] \leq O\left(\sqrt{\log KN}\sqrt{\frac{K}{N}}\right), \tag{70}$$

$$\mathbb{E}\left[\|f_t^N - \nabla F(\eta_t)\|^2\right] \leq O\left(\log(KN)\frac{K}{N}\right). \tag{71}$$

Below is the corrected theorem for the reward tracking error analysis and the accompanying proof.

Theorem 3. *Assume* $\gamma_t = (1+t)^{-\nu}, \alpha = \alpha_t'/\sqrt{\sum_{k=1}^t \|h_t\|^2}$, *and* $\alpha_t' = (1+t)^{-\sigma}$, *where* $0 < \nu < \sigma < 1$. *Then*

$$\frac{1}{T}\sum_{t=1}^{T} \mathbf{E}\left[(\eta_t - \eta_t^*)^2\right] \leq \mathcal{O}\left(T^{\nu-1}\right) + \mathcal{O}\left(T^{-2(\sigma-\nu)}\right) \tag{72}$$

$$+ \widetilde{\mathcal{O}}\left(\tau_{mix} \log T_{max}\right) \mathcal{O}\left(T^{-\nu}\right) \tag{73}$$

$$+ \widetilde{\mathcal{O}}\left(\tau_{mix} \frac{\log T_{max}}{T_{max}}\right). \tag{74}$$

Proof. Because the proof closely resembles the original version from (Suttle et al., 2023) with a few changes, we will only show intermediate steps for portions the changes affect. Similar to (Suttle et al., 2023), we recall that the average reward tracking update is given by

$$\eta_{t+1} = \eta_t - \gamma_t f_t, \tag{75}$$

where $f_t := f_t^{\text{MLMC}}$. We can rewrite the tracking error term $(\eta_{t+1} - \eta_{t+1}^*)^2$ as

$$\begin{aligned}(\eta_{t+1} - \eta_{t+1}^*)^2 \leq (1-2\gamma_t)(\eta_t - \eta_t^*)^2 + 2\gamma_t(\eta_t - \eta_t^*)(F'(\eta_t) - f_t) + 2(\eta_t - \eta_t^*)(\eta_t^* - \eta_{t+1}^*) \\ + 2(\eta_t^* - \eta_{t+1}^*)^2 + 2(\gamma_t f_t)^2.\end{aligned} \tag{76}$$

As in (Suttle et al., 2023), we take expectations and transform the expression into five separate summations,

$$\begin{aligned}\sum_{t=1}^{T} \mathbf{E}[(\eta_t - \eta_t^*)^2] \leq \underbrace{\sum_{t=1}^{T} \frac{1}{2\gamma_t} \mathbf{E}[(\eta_t - \eta_t^*)^2 - (\eta_t - \eta_t^*)^2]}_{I_1} + \underbrace{\sum_{t=1}^{T} \mathbf{E}[(\eta_t - \eta_t^*)(F'(\eta_t) - f_t)]}_{I_2} \\ + \underbrace{\sum_{t=1}^{T} \frac{1}{\gamma_t} \mathbf{E}[(\eta_t - \eta_t^*)(\eta_t^* - \eta_{t+1}^*)]}_{I_3} + \underbrace{\sum_{t=1}^{T} \frac{1}{\gamma_t} \mathbf{E}[(\eta_t^* - \eta_{t+1}^*)^2]}_{I_4} + \underbrace{\sum_{t=1}^{T} \gamma_t \mathbf{E}[(f_t)^2]}_{I_5}.\end{aligned} \tag{77}$$

(Suttle et al., 2023) provides bounds for I_1, I_2, I_3, I_4 and I_5. In this proof, only I_2 needs to be modified. So we will simply restate the bounds for the other terms and give more details for our modified I_2,

$$I_1 \leq \frac{r_{max}^2}{\gamma_T}, \tag{78}$$

where we use the fact that $(\eta_t - \eta_t^*)^2 \leq 2r_{max}^2$.

Bound on I_2: (Suttle et al., 2023) achieves this intermediate bound on the absolute value I_2.

$$|I_2| \leq \sum_{t=1}^{T} \mathbf{E}\left[\left|(\eta_t - \eta_t^*)\right| \cdot \left|(F'(\eta_t) - f_t^{j_{\max}})\right|\right]. \tag{79}$$

(Suttle et al., 2023) proceeds to bound $(\eta_t - \eta_t^*)^2 \leq 2r_{max}$. However, we will omit that step and bound in the term in the following way,

$$|I_2| \leq \sum_{t=1}^{T} \mathbf{E}\left[\left|(\eta_t - \eta_t^*)\right| \cdot \left|(F'(\eta_t) - f_t^{j_{\max}})\right|\right] \tag{80}$$

$$\leq \sum_{t=1}^{T} \mathbf{E}\left|(\eta_t - \eta_t^*)\right| \cdot \sum_{t=1}^{T} \mathbf{E}\left|(F'(\eta_t) - f_t^{j_{\max}})\right| \tag{81}$$

$$\leq \left(\sum_{t=1}^{T} \mathbf{E}\left[\left|(\eta_t - \eta_t^*)\right|^2\right]\right)^{\frac{1}{2}} \left(\sum_{t=1}^{T} \mathbf{E}\left[\left|(F'(\eta_t) - f_t^{j_{\max}})\right|^2\right]\right)^{\frac{1}{2}}. \tag{82}$$

For $\left(\sum_{t=1}^{T}\mathbb{E}\left[\left|(F'(\eta_t)-f_t^{j_{\max}})\right|^2\right]\right)^{\frac{1}{2}}$, we utilize Lemma 11 like Suttle et al. (2023) with $x_t=\eta_t, \nabla L(x_t)=\nabla F(\eta_t)$ and $l(x_t,z_t)=f_t$, and the fact that the Lipschitz constant of $\nabla F(\eta_t)$ is 1:

$$|I_2| \leq \left(\sum_{t=1}^{T}\mathbf{E}\left[\left|(\eta_t-\eta_t^*)\right|^2\right]\right)^{\frac{1}{2}} \widetilde{\mathcal{O}}\left(T\tau_{\text{mix}}\frac{\log T_{\max}}{T_{\max}}\right)^{\frac{1}{2}}. \tag{83}$$

Bound on I_3:

$$|I_3| \leq \left(\sum_{t=1}^{T}\mathbf{E}\left[(\eta_t-\eta_t^*)^2\right]\right)^{1/2}\left(L^2G_H^2\sum_{t=1}^{T}\frac{\alpha_t^2}{\gamma_t^2}\right)^{1/2}. \tag{84}$$

Bound on I_4:

$$I_4 \leq L^2G_H^2\sum_{t=1}^{T}\frac{\alpha^2}{\gamma_t}. \tag{85}$$

Bound on I_5:

$$I_5 \leq \sum_{t=1}^{T}\gamma_t\widetilde{\mathcal{O}}\left(R^2\tau_{\text{mix}}^{\theta_t}\log T_{\max}\right). \tag{86}$$

Combining the foregoing and recalling that $\gamma_t=(1+t)^{-\nu}, \alpha_t'=(1+t)^{-\sigma}$, $0<\nu<\sigma<1$, and $\alpha_t\leq\alpha_t'$, we get

$$\sum_{t=1}^{T}\mathbf{E}[(\eta_t-\eta_t^*)^2] \leq 2r_{\max}^2(1+T)^{\nu}+\left[L^2G_H^2+\widetilde{\mathcal{O}}\left(\tau_{\text{mix}}\log T_{\max}\right)\right]\sum_{t=1}^{T}(1+t)^{-\nu} \tag{87}$$

$$+\left(\sum_{t=1}^{T}\mathbf{E}\left[\left|(\eta_t-\eta_t^*)\right|^2\right]\right)^{\frac{1}{2}}\widetilde{\mathcal{O}}\left(T\tau_{\text{mix}}\frac{\log T_{\max}}{T_{\max}}\right)^{\frac{1}{2}} \tag{88}$$

$$+\left(\sum_{t=1}^{T}\mathbf{E}[(\eta_t-\eta_t^*)^2]\right)^{\frac{1}{2}}\left(L^2G_H^2\sum_{t=1}^{T}(1+t)^{-2(\sigma-\nu)}\right)^{\frac{1}{2}}, \tag{89}$$

where the second inequality follows from the fact that $\nu-2\sigma<-\nu$.

Define

$$Z(T)=\sum_{t=1}^{T}\mathbf{E}[(\eta_t-\eta_t^*)^2], \tag{90}$$

$$F(T)=\frac{L^2G_H^2}{4}\sum_{t=1}^{T}(1+t)^{-2(\sigma-\nu)}, \tag{91}$$

$$G(T)=T\widetilde{\mathcal{O}}\left(\tau_{\text{mix}}\frac{\log T_{\max}}{T_{\max}}\right), \tag{92}$$

$$A(T)=2r_{\max}^2(1+T)^{\nu}+\left[L^2G_H^2+\widetilde{\mathcal{O}}\left(\tau_{\text{mix}}\log T_{\max}\right)\right]\sum_{t=1}^{T}(1+t)^{-\nu}. \tag{93}$$

$$\tag{94}$$

The inequality can now be written as,

$$Z(T)\leq A(T)+2\sqrt{Z(T)}\sqrt{F(T)}+2\sqrt{Z(T)}\sqrt{G(T)}\leq 2A(T)+16F(T)+16G(T), \tag{95}$$

By following the same steps as the critic error analysis in (Suttle et al., 2023) to rearrange the above inequality, we achieve,

$$Z(T)\leq 2A(T)+16F(T)+16G(T). \tag{96}$$

From $2A(T)+16F(T) = \mathcal{O}\left(T^{\nu}\right)+\mathcal{O}\left(T^{1+\nu-2\sigma}\right)+\mathcal{O}\left(T^{1-\nu}\right)$ and using the bound $\sum_{t=1}^{T}(1+t)^{-\xi} \leq (1+t)^{1-\xi}/(1-\xi)$, we have by dividing by T,

$$\frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\left[(\eta_t-\eta_t^*)^2\right] \leq \mathcal{O}\left(T^{\nu-1}\right)+\mathcal{O}\left(T^{-2(\sigma-\nu)}\right)+\widetilde{\mathcal{O}}\left(\tau_{mix}\log T_{max}\right)\mathcal{O}\left(T^{-\nu}\right)+\widetilde{\mathcal{O}}\left(\tau_{mix}\frac{\log T_{max}}{T_{max}}\right). \tag{97}$$

□

D. Parameterized Policy Gradient with Advantage Estimation

We repeat the algorithm for PPGAE as it appears in (Bai et al., 2024).

Algorithm 1 Parameterized Policy Gradient

Input: Initial parameter θ_1, learning rate α, initial state $s_0 \sim \rho(\cdot)$, episode length H
$K = T/H$
for $k \in \{1, \cdots, K\}$ **do**
  $\mathcal{T}_k \leftarrow \phi$
  for $t \in \{(k-1)H, \cdots, kH-1\}$ **do**
    Execute $a_t \sim \pi_{\theta_k}(\cdot|s_t)$, receive reward $r(s_t, a_t)$ and observe s_{t+1}
    $\mathcal{T}_k \leftarrow \mathcal{T}_k \cup \{(s_t, a_t)\}$
  end for
  for $t \in \{(k-1)H, \cdots, kH-1\}$ **do**
    Using Algorithm 2, and $\mathcal{T}_k$, compute $\hat{A}^{\pi_{\theta_k}}(s_t, a_t)$
  end for
  Compute $\omega_k = \frac{1}{H}\sum_{t=t_k}^{t_{k+1}-1}\hat{A}^{\pi_\theta}(s_t, a_t)\nabla_\theta \log \pi_{\theta_k}(a_t|s_t)$
  Update parameters as
$$\theta_{k+1} = \theta_k + \alpha\omega_k \tag{98}$$
end for

Algorithm 2 Advantage Estimation

Input: Trajectory $(s_{t_1}, a_{t_1}, \ldots, s_{t_2}, a_{t_2})$, state s, action a, and policy parameter θ
Initialize: $i \leftarrow 0$, $\tau \leftarrow t_1$
Define: $N = 4t_{\text{mix}} \log_2 T$.
while $\tau \leq t_2 - N$ **do**
  if $s_\tau = s$ **then**
    $i \leftarrow i + 1$.
    $\tau_i \leftarrow \tau$
    $y_i = \sum_{t=\tau}^{\tau+N-1} r(s_t, a_t)$.
    $\tau \leftarrow \tau + 2N$.
  else
    $\tau \leftarrow \tau + 1$.
  end if
end while
if $i > 0$ **then**
  $\hat{V}(s) = \frac{1}{i} \sum_{j=1}^{i} y_j$,
  $\hat{Q}(s,a) = \frac{1}{\pi_\theta(a|s)} \left[\frac{1}{i} \sum_{j=1}^{i} y_j 1(a_{\tau_j} = a) \right]$
else
  $\hat{V}(s) = 0$, $\hat{Q}(s,a) = 0$
end if
return $\hat{Q}(s,a) - \hat{V}(s)$

E. Multi-level Actor-Critic

We repeat the algorithm overview for MAC as it appears in (Suttle et al., 2023).

Algorithm 3 **M**ulti-level Monte Carlo **A**ctor-**C**ritic (MAC)

Initialize: Policy parameter θ_0, actor step size α_t, critic step size β_t, average reward tracking step size γ_t, initial state $s_1^{(0)} \sim \mu_0(\cdot)$, maximum trajectory length $T_{\max}$.
for $t = 0$ **to** $T - 1$ **do**
  Sample level length $j_t \sim \text{Geom}(1/2)$
  for $i = 1, \ldots, 2^{j_t}$ **do**
    Take action $a_t^i \sim \pi_{\theta_t}(\cdot|s_t^i)$
    Collect next state $s_t^{i+1} \sim P(\cdot|s_t^i, a_t^i)$
    Receive reward $r_t^i = r(s_t^i, a_t^i)$
  end for
  Evaluate MLMC gradient f_t^{MLMC}, h_t^{MLMC}, and g_t^{MLMC} via (17)
  Update parameters following (15)
end for