Global Optimality without Mixing Time Oracles in Average-reward RL via Multi-level Actor-Critic

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

[1]University of Maryland, College Park
[2]U.S. Army Research Laboratory, MD, USA
[3]JP Morgan AI Research, NYC
[4]Purdue University
[5]University of Central Florida

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 $\mathcal{O}\left(\sqrt{\tau_{mix}}\right)$ relative to prior work. With a 2D gridworld goal-reaching navigation experiment, we demonstrate the MAC achieves higher reward than a previous PG-based method for average reward Parameterized Policy Gradient with Advantage Estimation (PPGAE), especially in cases with relatively small training sample budget restricting trajectory length.

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; Agarwal et al., 2020; Mei et al., 2020; Liu et al., 2020; Bedi et al., 2022). 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, leaving 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 the 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 the end paper with conclusion and discussion on future work.

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

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. Its global optimality has been shown to exist for softmax (Mei et al., 2020) and tabular (Bhandari & Russo, 2024; Agarwal et al., 2020; Kumar et al., 2024) 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. Recently, Bai et al. (2024) adapted this framework for the average reward infinite horizon MDP with general policy parameterization. We wish to 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 Suttle et al. (2023), they relaxed this assumption with their proposed Multi-level Actor-Critic (MAC) while still recovering SOTA convergence. In this paper, we aim to establish the global convergence of MAC and highlight its tighter dependence on mixing time despite no oracle knowledge.

3 Problem Formulation

3.1 Average Reward Policy Optimization

Reinforcement learning problem with the average reward criterion may be formalized as a Markov decision process $\mathcal{M} := (\mathcal{S}, \mathcal{A}, \mathbb{P}, r)$. In this tuple expression, $\mathcal{S}$ 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 $s'_t \sim \mathbb{P}(\cdot \mid s_t, a_t)$ and the environment reveals a reward $r(s_t, a_t)$. Actions may be selected according to a policy $\pi(\cdot \mid s)$, which is a distribution over action space $\mathcal{A}$ given current state s.

We aim to maximize the long-term average reward $J(\pi) := \lim_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T} r(s_t, a_t)\right]$ by finding the optimal policy π. The policy is parameterized by vector $\theta \in \mathbb{R}^q$, where q denotes the parameter dimension, and the policy dependence on θ is 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:

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

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). Further define the stationary distribution of induced by a parameterized policy as,

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

As we will later discuss, the induced d^{π_θ} for a given θ is unique and therefore, is agnostic to initial distribution ρ, due to ergodic assumption. We are now able to express the average reward with respect 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 thus 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-out samples and lead to noisy gradients. Thus, knowing the time it takes an induced Markov Chain has reached its stationary distribution is a crucial element for policy gradient algorithms. The quantity is known as *mixing time* τ_{mix} as is defined as,

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

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

where $\|\cdot\|_{TV}$ is the total variation distance.

The ϵ-mixing time of a Markov chain is defined as

$$\tau^{\theta}_{mix}(\epsilon) := \inf\{t : m(t;\theta) \le \epsilon\}, \tag{4}$$

Typical in PG analysis, mixing time is defined as $\tau^{\theta}_{mix} := \tau^{\theta}_{mix}(1/4)$ as it provides a result in that $m(l\tau^{\theta}_{mix};\theta) \le 2^{-l}\}$ Dorfman & Levy (2022). We further define $\tau_{mix} = \max_{t\in[T]} \tau^{\theta_t}_{mix}$.

In prior works such as Duchi et al. (2012); Nagaraj et al. (2020); Bai et al. (2024); Wei et al. (2021), τ^{θ}_{mix} 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, however, briefly introduce the elements of a vanilla actor-critic in Section 3.3.

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

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

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

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

Using the Bellman's Equation, we can write, from (5) and (6), the value of a state s, in terms of another as (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 term as follows,

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

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

Lemma 2. *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,

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

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 epochs, H as the length of one epoch, 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 epoch, the advantage value of each sample collected in an epoch 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:

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

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, by definition H, even if mixing time and hitting time are known, the minimum T for $K > 1$ 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 $V_\omega(s) \coloneqq \langle \phi(s), \omega \rangle$ 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

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

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 with 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

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

where we have

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

As the critic and reward tracking are vital to the 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. That is for some $\rho \in [0, 1], m(t; \theta) \leq \mathcal{O}\left(\rho^t\right)$. However most vanilla AC analysis assumes there exists some ρ such that for all $\theta, m(t; \theta) \leq \mathcal{O}\left(\rho^t\right)$. This assumption sets an upper limit on how slow mixing an environment can be for the algorithm to handle. In the following section, we explain the AC variant, MAC, that will provide with the tighter dependence on mixing time despite no oracle knowledge of it and 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 $\mathcal{T}_t := \{s_t^i, a_t^i, r_t^i, s_t^{i+1}\}_{i=1}^{2^{K_t}}$ 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 8, 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) utilized the adpative stepsize scheme, Adagrad (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 support our global convergence analysis in the next section.

Assumption 3. *For all θ, the parameterized MDP $\mathcal{M}_\subseteq$ induces an ergodic Markov Chain.*

Assumption 3 is typical in many works such as Suttle et al. (2023); Pesquerel & Maillard (2022); Gong & Wang (2020); Bai et al. (2024). As previously mentions, it ensures all states are reachable, and also importantly, guarantees a unique stationary distribution d^{π_θ} of any induced Markov Chain.

Because we parameterize the critic as a linear function approximator, 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 4. *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 5. *Let $\{\pi_\theta\}_{\theta\in\mathbb{R}^d}$ denote our parameterized policy class. There exist $B, K, L > 0$ 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 5 establishes regularization conditions for the policy gradient ascent and has been utilized in prior work such as 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 will be vital for presenting our modified general framework for non-constant stepsize in Lemma 12 as B, R, L will appear in our bound for the difference between optimal reward and the average cumulative reward.

Assumption 6. *Define the transferred policy function approximation error*

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

where π^* is the optimal policy and h_θ^* is given as

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

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

Assumption 6 bounds the error that arises from the policy class parameterization. For neural networks, $\mathcal{E}_{app}^{actor}$ has 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 we will later see in Section 4 as $\mathcal{E}_{app}^{actor}$ will appear as its own independent term in the final bound.

For our later analysis, we also define

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

as the Fisher information matrix. 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 operation.

Assumption 7. *Setting I_F as the identity matrix of same dimensionality as $F(\theta)$, let there exists some positive constant μ_F such that $F(\theta) - \mu_F I_F$ is positive semidefinite.*

Assumption 7 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 12 into terms of $\left[\left|\left|\nabla J(\theta_t)\right|\right|^2\right]$, which MAC has an bound for established by Suttle et al. (2023). We provide that bound in Lemma 10.

Lemma 8. *Let $j_{max} = \lfloor \log T_{max} \rfloor$. Fix θ_t measurable w.r.t. $\mathcal{F}_{t-1}$. Assume $T_{max} \geq \tau_{mix}^{\theta_t}$, $||\nabla J(\theta)|| \leq G_H$, for all θ, and $||h_t^N|| \leq G_H$, for all $N \in [T_{max}]$. Then*

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

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

Lemma 8 provides a bound for the variance of the MLMC gradient estimator and will integral to our analysis.

Lemma 9. *Assume $\gamma_t = (1+t)^{-\nu}, \alpha = \alpha_t'/\sqrt{\sum_{k=1}^t ||h_t^{MLMC}||^2}$, and $\alpha_t' = (1+t)^{-\sigma}$, where $0 < \nu < \sigma < 1$. Then*

$$\frac{1}{T}\sum_{t=1}^T \mathcal{E}(t) \leq \mathcal{O}\left(T^{\nu-1}\right) + \mathcal{O}\left(T^{-2(\sigma-\nu)}\right) + \widetilde{\mathcal{O}}\left(\tau_{mix}\log T_{\max}\right)\mathcal{O}\left(T^{-\nu}\right) + \widetilde{\mathcal{O}}\left(\tau_{mix}\frac{\log T_{\max}}{T_{\max}}\right). \tag{22}$$

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

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

Lemma 9 established by Suttle et al. (2023) states the convergence rate of the critic with an MLMC estimator. This result directly affects the overall MAC convergence rate from Suttle et al. (2023) stated below,

Lemma 10. *(MAC Convergence Rate) 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 9, we have*

$$\frac{1}{T}\sum_{t=1}^T \mathbb{E}\left[\left|\left|\nabla J(\theta_t)\right|\right|^2\right] \leq \mathcal{O}\left(\mathcal{E}_{app}^{critic}\right) + \widetilde{\mathcal{O}}\left(\frac{\tau_{mix}\log T_{\max}}{\sqrt{T}}\right) + \widetilde{\mathcal{O}}\left(\frac{\tau_{mix}\log T_{\max}}{T_{\max}}\right). \tag{23}$$

Both Lemmas 9 and 10 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 58. We provide a correct version of the proof of Lemma D.1 in the Appendix where we able to change remove the square root.

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

Lemma 11. *Lemma 4.2, Dorfman & Levy (2022). For any non-negative real numbers* $\{a_i\}_{i\in[n]}$,

$$\sum_{i=1}^{n} \frac{a_i}{\sqrt{\sum_{j=1}^{i} a_j}} \leq 2\sqrt{\sum_{i=1}^{n} a_i}. \tag{24}$$

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 Adagrad.

Lemma 12. *Suppose a general gradient ascent algorithm updates the policy parameter in the following way.*

$$\theta_{t+1} = \theta_t + \alpha_t h_t. \tag{25}$$

When Assumptions 5, 6, and 7 hold, we have the following inequality for any T.

$$J^* - \frac{1}{T}\sum_{t=1}^{T} J(\theta_t) \leq \sqrt{\mathcal{E}_{app}^{actor}} + \frac{B}{T}\sum_{t}^{T}\|(h_t - h_t^*)\| + \frac{R}{2T}\sum_{t=1}^{T}\alpha_t\|h_t\|^2 + \frac{1}{T}\sum_{t=1}^{T}\frac{1}{\alpha_t}\mathbf{E}_{s\sim d^{\pi^*}}\zeta_t, \tag{26}$$

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* $\zeta_t = [KL(\pi^*(\cdot|s)\|\pi_{\theta_k}(\cdot|s)) - KL(\pi^*(\cdot|s)\|\pi_{\theta_{k+1}}(\cdot|s))]$.

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 without bounding α_t to some constant which we will do later in the analysis.

Theorem 13. *Let* $\{\theta_t\}_{t=1}^{T}$ *be defined as in Lemma 12. If assumptions 3, 5, 6, 7 hold,* $J(\cdot)$ *is* L*-smooth, then the following inequality holds.*

$$J^* - \frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\left[J(\theta_t)\right] \leq \sqrt{\mathcal{E}_{app}^{actor}} + \widetilde{\mathcal{O}}\left(\frac{\sqrt{\tau_{mix}T_{\max}}\log T_{\max}}{T^{\frac{1}{8}}}\right) + \mathcal{O}\left(\frac{\sqrt{\log(T_{max})T_{max}\mathcal{E}_{app}^{critic}}}{\sqrt{T}}\right) + \widetilde{\mathcal{O}}\left(\frac{\sqrt{\tau_{mix}}\log T_{\max}}{\sqrt{T_{\max}}}\right). \tag{27}$$

The proof of this above theorem can be found in Appendix B. Here, we provide a proof sketch to highlight the main mechanics.

Proof sketch. We rewrite the bound of the expectation of 26 of Lemma 12 into terms of 22 of Lemma 8 and 23 of Lemma 10.

The expectation of the second term of the RHS of 26 can be bounded as the following using Assumption 7,

$$\frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\|h_t - h_t^*\| \leq \sqrt{\frac{2}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|h_t\|^2\right]} + \sqrt{\frac{2}{T}\sum_{t=1}^{T}\left(2 + \frac{1}{\mu_F^2}\right)\mathbf{E}\left[\|\nabla_\theta J(\theta_t)\|^2\right]}. \tag{28}$$

For the third term of the RHS of 26, we can utilize Lemma 11

$$\frac{R}{2T}\sum_{t=1}^{T}\alpha_t\|h_t\|^2 \leq \frac{R}{T}\sqrt{\sum_{t=1}^{T}\|h_t\|^2}. \tag{29}$$

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

$$\frac{1}{T}\sum_{t=1}^{T}\frac{1}{\alpha_t}\mathbf{E}_{s\sim d^{\pi^*}}[\zeta_t] \leq \frac{\mathbf{E}_{s\sim d^{\pi^*}}[KL(\pi^*(\cdot|s)\|\pi_{\theta_1}(\cdot|s))]\sqrt{\sum_{t=1}^{T}\|h_t\|^2}}{T\alpha'_T}. \tag{30}$$

Plugging in these bounds back into 26 and ignoring constants:

$$J^* - \frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\|J(\theta_t)\| \leq \sqrt{\mathcal{E}_{app}^{actor}} + \mathcal{O}\left(\frac{1}{\sqrt{T}}\right)\sqrt{\frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|h_t\|^2\right]} + \sqrt{\frac{2}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|\nabla_\theta J(\theta_t)\|^2\right]}. \tag{31}$$

Bounding the the second and third term by the RHS with Lemmas 8 and 10 respectively concludes the proof.

Remark. 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 $\mathcal{E}_{app}^{actor} \geq 0$ term accounts for the general policy parameterization. However, our bound has no dependence on hitting time like Bai et al. (2024) as their dependence was a result of their advantage estimation algorithm described in Section 3.2.

Our final theoretical result follows from setting the critic estimation error $\mathcal{E}(t)$ and critic approximation error $\mathcal{E}_{app}^{critic}$ both to 0.

Corollary 14. *With the same assumptions as Theorem 13 and by setting* $\mathcal{E}(t) = \mathcal{E}_{app}^{critic} = 0$*,*

$$J^* - \frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\|J(\theta_t)\| \leq \sqrt{\mathcal{E}_{app}^{actor}} + \widetilde{\mathcal{O}}\left(\sqrt{\frac{\tau_{mix}\log T_{\max}}{T_{\max}}}\right) + \widetilde{\mathcal{O}}\left(\sqrt{\frac{\tau_{mix}\log T_{max}}{T}}\right) + \mathcal{O}\left(T^{-\frac{1}{4}}\right). \tag{32}$$

The proof of Corollary 14 is a simplified version of the proof of Theorem 13 presented here where all terms that arise from the critic error in the supporting lemmas vanish.

Remark. With Corollary 14, we can recover the $\mathcal{O}\left(T^{-\frac{1}{4}}\right)$ as in Bai et al. (2024).

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 epochs, and $H = 16\tau_{hit}\tau_{mix}\sqrt{T}(\log(T))^2$ as the length of one epoch, thus $K = T/H$. Because K represents the number of epochs, it must be a positive integer. For $K = T/H \leq 1$, that 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 $\tau_{hit} = 10$, 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.

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. The expected number of samples per trajectory is $\mathbf{E}\left[2^j\right] = 4$ as $j \sim \text{Geom}(\frac{1}{2})$. Thus our algorithm has a much more practical minimum number of samples needed to run.

4.3 Low Budget Training

As a preliminary proof-of-concept experiment to show the advantage of MAC over PPGAE with a relatively small training sample budget. We consider a 5-by-5 sparse gridworld environment. The

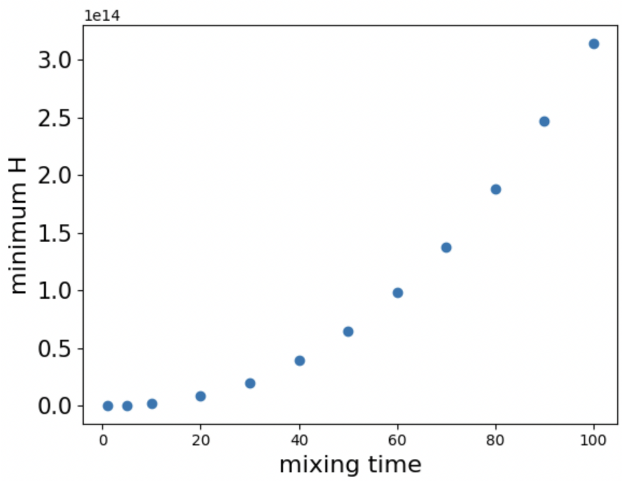

Figure 1: Minimum H required for $K = 1$ given a mixing time τ_{mix}. Both H and $\tau_{\max}$ are in terms of number of samples. We set the hitting time to be 10 for this plot.

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 25 samples. We report a moving average success rate over 5 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 PPGAE in the few number of steps per episode.

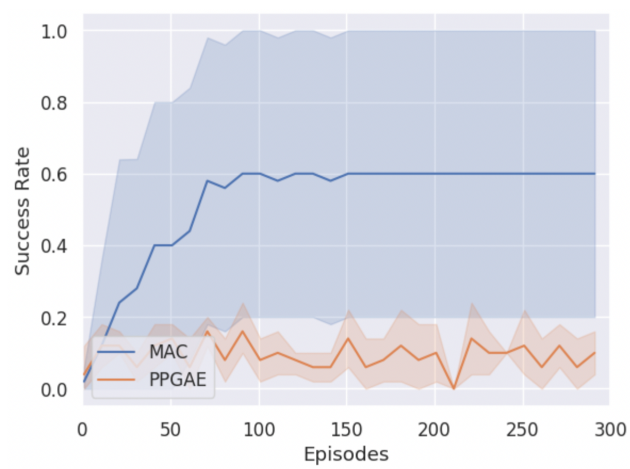

Figure 2: Success Rate in a sparse 5-by-5 grid over 300 training episodes with 25 samples per episode. For MAC, $T_{max} = 4$ and for PPGAE, $H = 25$ and $N = 1$. 5 trials for each algorithm. With a low-number of a samples for training, PPGAE convergences to significantly less optimal solution than MAC, highlighting the superiority in sample efficiency of 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.

Acknowledgements

This research was supported by Army Cooperative Agreement W911NF2120076 and ARO Grant W911NF2310352

Disclaimer: This paper was prepared for informational purposes in part by the Artificial Intelligence Research group of JPMorgan Chase & Coand 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.

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.

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.

Kumar, N., Murthy, Y., Shufaro, I., Levy, K. Y., Srikant, R., and Mannor, S. On the global convergence of policy gradient in average reward markov decision processes, 2024.

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.

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.

Appendix

A Proof of Lemma 12

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

$$J(\theta) - J(\theta') = \mathbf{E}_{s\sim d^{\pi_\theta}}\mathbf{E}_{a\sim\pi_\theta(\cdot|s)}[A^{\pi_{\theta'}}(s,a)] \tag{33}$$

We can now provide the general framework lemma.

Lemma 16. *Suppose a general gradient ascent algorithm updates the policy parameter in the following way.*

$$\theta_{t+1} = \theta_t + \alpha_t h_t \tag{34}$$

When Assumptions 5, 6, and 15 hold, we have the following inequality for any T.

$$\begin{aligned} J^* - \frac{1}{T}\sum_{t=1}^{T} J(\theta_t) \leq \sqrt{\mathcal{E}_{app}^{actor}} + \frac{B}{T}\sum_{t}^{T}\|(h_t - h_t^*)\| \\ + \frac{K}{2T}\sum_{t=1}^{T}\alpha_t\|h_t\|^2 + \frac{1}{T}\sum_{t=1}^{T}\frac{1}{\alpha_t}\mathbf{E}_{s\sim d^{\pi^*}}\zeta_t \end{aligned} \tag{35}$$

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 $\pi_{\theta+1}$.

$$
\begin{aligned}
&\mathbf{E}_{s\sim d^{\pi^*}}[KL(\pi^*(\cdot|s)\|\pi_{\theta_t}(\cdot|s)) - KL(\pi^*(\cdot|s)\|\pi_{\theta_{t+1}}(\cdot|s))] \\
&= \mathbf{E}_{s\sim d^{\pi^*}}\mathbf{E}_{a\sim\pi^*(\cdot|s)}\left[\log\frac{\pi_{\theta_{t+1}(a|s)}}{\pi_{\theta_t}(a|s)}\right] \\
&\overset{(a)}{\geq} \mathbf{E}_{s\sim d^{\pi^*}}\mathbf{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 \\
&= \mathbf{E}_{s\sim d^{\pi^*}}\mathbf{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 \\
&= \mathbf{E}_{s\sim d^{\pi^*}}\mathbf{E}_{a\sim\pi^*(\cdot|s)}[\nabla_\theta \log\pi_{\theta_t}(a|s)\cdot\alpha_t h_t^*] + \mathbf{E}_{s\sim d^{\pi^*}}\mathbf{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 \\
&= \alpha_t[J^*-J(\theta_t)] + \mathbf{E}_{s\sim d^{\pi^*}}\mathbf{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)] \\
&+ \mathbf{E}_{s\sim d^{\pi^*}}\mathbf{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 \\
&\overset{(b)}{=} \alpha_t[J^*-J(\theta_t)] + \alpha_t\mathbf{E}_{s\sim d^{\pi^*}}\mathbf{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] \\
&+ \alpha_t\mathbf{E}_{s\sim d^{\pi^*}}\mathbf{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 \\
&\overset{(c)}{\geq} \alpha_t[J^*-J(\theta_t)] - \alpha_t\sqrt{\mathbf{E}_{s\sim d^{\pi^*}}\mathbf{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]} \\
&- \alpha_t\mathbf{E}_{s\sim d^{\pi^*}}\mathbf{E}_{a\sim\pi^*(\cdot|s)}\|\nabla_\theta \log\pi_{\theta_t}(a|s)\|_2\|(h_t-h_t^*)\| - \frac{K\alpha_t^2}{2}\|h_t\|^2 \\
&\overset{(d)}{\geq} \alpha_t[J^*-J(\theta_t)] - \alpha_t\sqrt{\mathcal{E}_{app}^{actor}} - \alpha_t B\|(h_t-h_t^*)\| - \frac{K\alpha_t^2}{2}\|h_t\|^2
\end{aligned}
\tag{36}
$$

where we use Assumption 5 for step (a) and Lemma 15 for step (b). Step (c) uses the convexity of the function $f(x) = x^2$, and (d) comes from Assumption 6. We can get by rearranging terms,

$$
\begin{aligned}
J^* - J(\theta_t) \leq \sqrt{\mathcal{E}_{app}^{actor}} + B\|(h_t-h_t^*)\| + \frac{K\alpha_t}{2}\|h_t\|^2 \\
+ \frac{1}{\alpha_t}\mathbf{E}_{s\sim d^{\pi^*}}[KL(\pi^*(\cdot|s)\|\pi_{\theta_t}(\cdot|s)) - KL(\pi^*(\cdot|s)\|\pi_{\theta_{t+1}}(\cdot|s))]
\end{aligned}
\tag{37}
$$

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

B Proof of Theorem 13

To use 26 for our convergence analysis, we will take the expectation of the second term. With $h_t^{MLMC} = h_t$, note that,

$$\begin{aligned}
\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\|h_t - h_t^*\|\right)^2 &\leq \frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|h_t - h_t^*\|^2\right] \\
&= \frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|h_t - F(\theta_t)^\dagger\nabla_\theta J(\theta_t)\|^2\right] \\
&\leq \frac{2}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|h_t - \nabla_\theta J(\theta_t)\|^2\right] \\
&\quad + \frac{2}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|\nabla_\theta J(\theta_t) - F(\theta_t)^\dagger\nabla_\theta J(\theta_t)\|^2\right] \\
&\overset{(a)}{\leq} \frac{2}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|h_t - \nabla_\theta J(\theta_t)\|^2\right] \\
&\quad + \frac{2}{T}\sum_{t=1}^{T}\left(1 + \frac{1}{\mu_F^2}\right)\mathbf{E}\left[\|\nabla_\theta J(\theta_t)\|^2\right]
\end{aligned} \tag{38}$$

where (a) uses Assumption 7.

We can further break down the first term of the RHS:

$$\begin{aligned}
\frac{2}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|h_t - \nabla_\theta J(\theta_t)\|^2\right] \leq \frac{2}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|h_t\|^2\right] \\
+\frac{2}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|\nabla_\theta J(\theta_t)\|^2\right]
\end{aligned} \tag{39}$$

Plugging inequality 39 back to the RHS and combining like terms:

$$\begin{aligned}
\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\|h_t - h_t^*\|\right)^2 \leq \frac{2}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|h_t\|^2\right] \\
+\frac{2}{T}\sum_{t=1}^{T}\left(2 + \frac{1}{\mu_F^2}\right)\mathbf{E}\left[\|\nabla_\theta J(\theta_t)\|^2\right]
\end{aligned} \tag{40}$$

Taking the square root of both sides and from the fact $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$:

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\|h_t - h_t^*\| \leq \sqrt{\frac{2}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|h_t\|^2\right]} \\
+\sqrt{\frac{2}{T}\sum_{t=1}^{T}\left(2 + \frac{1}{\mu_F^2}\right)\mathbf{E}\left[\|\nabla_\theta J(\theta_t)\|^2\right]}
\end{aligned} \tag{41}$$

We can also bound the third term of the RHS of 26 with Lemma 11

$$\frac{R}{2T}\sum_{t=1}^{T}\alpha_t\|h_t\|^2 \leq \frac{R}{2T}\sum_{t=1}^{T}\frac{\|h_t\|^2}{\sqrt{\sum_{o=1}^{t}\|h_o\|^2}} \leq \frac{R}{T}\sqrt{\sum_{t=1}^{T}\|h_t\|^2} \tag{42}$$

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

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\frac{1}{\alpha_t}\mathbf{E}_{s\sim d^{\pi^*}}[\zeta_t] &\leq \frac{1}{T}\sum_{t=1}^{T}\frac{\mathbf{E}_{s\sim d^{\pi^*}}[\zeta_t]}{\alpha_T}\\ &= \frac{1}{T}\frac{\mathbf{E}_{s\sim d^{\pi^*}}[KL(\pi^*(\cdot|s)\|\pi_{\theta_1}(\cdot|s))]}{\alpha_T}\\ &\leq \frac{\mathbf{E}_{s\sim d^{\pi^*}}[KL(\pi^*(\cdot|s)\|\pi_{\theta_1}(\cdot|s))]\sqrt{\sum_{t=1}^{T}\|h_t\|^2}}{T\alpha'_T}\end{aligned} \tag{43}$$

Taking the expectation of both sides of 26 and plugging in 41, 42, and 30, ignoring constants:

$$\begin{aligned}&J^* - \frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\|J(\theta_t)\| \leq \sqrt{\mathcal{E}_{app}^{actor}}\\ &+\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)\sqrt{\frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|h_t\|^2\right]}+\sqrt{\frac{2}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|\nabla_\theta J(\theta_t)\|^2\right]}\end{aligned} \tag{44}$$

From Lemma 8 we can bound the summation of the expected variance of the MLMC gradient. Ignoring the G_H constant,

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|h_t\|^2\right] \leq &\frac{1}{T}\sum_{t=1}^{T}\widetilde{\mathcal{O}}\left(\tau_{mix}^{\theta_t}\log T_{max}\right)\\ &+\frac{1}{T}\sum_{t=1}^{T}\log(T_{max})T_{max}\mathcal{E}_2(t)\\ &+\frac{1}{T}\sum_{t=1}^{T}\log(T_{max})T_{max}(\mathcal{E}_{app}^{critic})^2\end{aligned} \tag{45}$$

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

$$\frac{1}{T}\sum_{t=1}^{T}\widetilde{\mathcal{O}}\left(\tau_{mix}^{\theta_t}\log T_{max}\right) \leq \widetilde{\mathcal{O}}\left(\tau_{mix}\log T_{max}\right) \tag{46}$$

For the second term by using 23,

$$\begin{aligned}&\frac{1}{T}\sum_{t=1}^{T}\log(T_{max})T_{max}\mathcal{E}(t)\\ &\quad\leq (\log T_{\max})T_{\max}\widetilde{\mathcal{O}}\left(\tau_{mix}(\log T_{\max})\right)\mathcal{O}\left(T^{-\frac{1}{2}}\right)\\ &\quad+(\log T_{\max})T_{\max}\widetilde{\mathcal{O}}\left(\tau_{mix}\frac{(\log T_{\max})}{T_{\max}}\right)\\ &\quad=\widetilde{\mathcal{O}}\left(\frac{\tau_{mix}(\log T_{\max})^2T_{\max}}{T^{\frac{1}{2}}}\right).\end{aligned} \tag{47}$$

The third term can be simply bounded as as follows:

$$\frac{1}{T}\sum_{t=1}^{T}\log(T_{max})T_{max}(\mathcal{E}_{app}^{critic})^2 \leq \mathcal{O}\left(\log(T_{max})T_{max}(\mathcal{E}_{app}^{critic})^2\right). \tag{48}$$

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

$$\frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|h_t\|^2\right] \leq \widetilde{\mathcal{O}}\left(\frac{\tau_{mix}(\log T_{\max})^2 T_{\max}}{T^{\frac{1}{2}}}\right) + \mathcal{O}\left(\log(T_{max})T_{max}(\mathcal{E}_{app}^{critic})^2\right) \tag{49}$$

Taking the square root of both sides, from $\sqrt{a+b} \leq \sqrt{a}+\sqrt{b}$, and multiplying by $\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$,

$$\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)\sqrt{\frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|h_t\|^2\right]} \leq \widetilde{\mathcal{O}}\left(\frac{\sqrt{\tau_{mix}T_{\max}}\log T_{\max}}{T^{\frac{1}{8}}}\right) + \mathcal{O}\left(\frac{\sqrt{\log(T_{max})T_{max}}\mathcal{E}_{app}^{critic}}{\sqrt{T}}\right) \tag{50}$$

From Lemma 10 we can bound the square root of the summation of the expectation of the gradient norm squared:

$$\sqrt{\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left|\left|\nabla J(\theta_t)\right|\right|^2\right]} \leq \mathcal{O}\left(\sqrt{\mathcal{E}_{app}^{critic}}\right) + \widetilde{\mathcal{O}}\left(\frac{\sqrt{\tau_{mix}\log T_{\max}}}{T^{\frac{1}{4}}}\right) + \widetilde{\mathcal{O}}\left(\frac{\sqrt{\tau_{mix}\log T_{\max}}}{\sqrt{T_{\max}}}\right) \tag{51}$$

Combining (50) and (51) we can get the final global convergence.

C Proof of Corollary 14

To prove Corollary 14 we restate Theorem 1 from Suttle et al. (2023) that explicitly characterizes the convergence rate of the multi-level actor in terms of critic estimation error, $\mathcal{E}(t)$, and critic approximation error, $\mathcal{E}_{app}^{critic}$. We present the theorem in the setting where $\mathcal{E}(t) = \mathcal{E}_{app}^{critic} = 0$.

Theorem 17. *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. Let $\alpha_t = \alpha'_t/\sqrt{\sum_{t=1}^{T}||h_t^{MLMC}||^2}$, where $\{\alpha'_t\}$ is an auxiliary stepsize sequence with $\alpha'_t \leq 1$, for all $t \geq 1$. Then*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left|\left|\nabla J(\theta_t)\right|\right|^2\right] \leq \mathcal{O}\left(\frac{1}{\sqrt{T}}\right) + \widetilde{\mathcal{O}}\left(\tau_{mix}\frac{\log T_{\max}}{T_{\max}}\right), \tag{52}$$

We will first restate Corollary 14 for convenience and then provide the proof.

Corollary 18. *With the same assumptions as Theorem 13 and by setting $\mathcal{E}(t) = \mathcal{E}_{app}^{critic} = 0$,*

$$J^* - \frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\|J(\theta_t)\| \leq \sqrt{\mathcal{E}_{app}^{actor}} + \widetilde{\mathcal{O}}\left(\sqrt{\frac{\tau_{mix}\log T_{\max}}{T_{\max}}}\right) + \widetilde{\mathcal{O}}\left(\sqrt{\frac{\tau_{mix}\log T_{max}}{T}}\right) + \mathcal{O}\left(T^{-\frac{1}{4}}\right). \tag{53}$$

Proof. We start the proof with Equation 31,

$$J^* - \frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\|J(\theta_t)\| \leq \sqrt{\mathcal{E}_{app}^{actor}} + \mathcal{O}\left(\frac{1}{\sqrt{T}}\right)\sqrt{\frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|h_t\|^2\right]} + \sqrt{\frac{2}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|\nabla_\theta J(\theta_t)\|^2\right]} \tag{54}$$

We can bound the second term of the RHS with Lemma 8 as we did for Theorem 13 but with the added assumption of $\mathcal{E}(t) = \mathcal{E}_{app}^{critic} = 0$:

$$\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)\sqrt{\frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|h_t\|^2\right]} \leq \mathcal{O}\left(\frac{1}{\sqrt{T}}\right)\sqrt{\frac{1}{T}\sum_{t=1}^{T}\widetilde{\mathcal{O}}\left(\tau_{mix}^{\theta_t}\log T_{max}\right)} \leq \widetilde{\mathcal{O}}\left(\sqrt{\frac{\tau_{mix}\log T_{max}}{T}}\right) \tag{55}$$

To bound the the third term of 31, we utilize Theorem 17,

$$\sqrt{\frac{2}{T}\sum_{t=1}^{T}\mathbf{E}\left[\|\nabla_\theta J(\theta_t)\|^2\right]} \leq \mathcal{O}\left(T^{-\frac{1}{4}}\right) + \widetilde{\mathcal{O}}\left(\sqrt{\frac{\tau_{mix}\log T_{\max}}{T_{\max}}}\right) \tag{56}$$

□

Plugging 55 and 56 back into Equation 31 we complete the proof,

$$J^* - \frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\|J(\theta_t)\| \leq \sqrt{\mathcal{E}_{app}^{actor}} + \widetilde{\mathcal{O}}\left(\sqrt{\frac{\tau_{mix}\log T_{max}}{T}}\right) + \mathcal{O}\left(T^{-\frac{1}{4}}\right) + \widetilde{\mathcal{O}}\left(\sqrt{\frac{\tau_{mix}\log T_{\max}}{T_{\max}}}\right) \tag{57}$$

D Corrected Analysis of Multi-level Monte Carlo

In this section, we wish to provide a corrected analysis for Lemma 10. 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 10. We then provide a corrected version of the average reward tracking analysis.

D.1 Overview of Correction

We repeat Lemma 10,

Lemma 19. *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 9, we have*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left|\left|\nabla J(\theta_t)\right|\right|^2\right] \leq \mathcal{O}\left(\mathcal{E}_{app}^{critic}\right) + \widetilde{\mathcal{O}}\left(\frac{\tau_{mix}\log T_{\max}}{\sqrt{T}}\right) + \widetilde{\mathcal{O}}\left(\frac{\tau_{mix}\log T_{\max}}{T_{\max}}\right). \tag{58}$$

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 9, the convergence of the critic estimation $\mathcal{E}(t)$, which we repeat here,

Lemma 20. *Assume $\gamma_t = (1+t)^{-\nu}, \alpha = \alpha'_t/\sqrt{\sum_{k=1}^{t}||h_t^{MLMC}||^2}$, and $\alpha'_t = (1+t)^{-\sigma}$, where $0 < \nu < \sigma < 1$. Then*

$$\frac{1}{T}\sum_{t=1}^{T}\mathcal{E}(t) \leq \mathcal{O}\left(T^{\nu-1}\right) + \mathcal{O}\left(T^{-2(\sigma-\nu)}\right) + \widetilde{\mathcal{O}}\left(\tau_{mix}\log T_{\max}\right)\mathcal{O}\left(T^{-\nu}\right) + \widetilde{\mathcal{O}}\left(\tau_{mix}\frac{\log T_{\max}}{T_{\max}}\right). \tag{59}$$

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 9 is dependent on the average reward tracking error. We repeat its convergence theorem from Suttle et al. (2023) below,

Theorem 21. *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*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\left[(\eta_t - \eta_t^*)^2\right] \leq \mathcal{O}\left(T^{\nu-1}\right) + \mathcal{O}\left(T^{-2(\sigma-\nu)}\right) \tag{60}$$

$$+\widetilde{\mathcal{O}}\left(\tau_{mix}\log T_{max}\right)\mathcal{O}\left(T^{-\nu}\right) \tag{61}$$

$$+\widetilde{\mathcal{O}}\left(\sqrt{\tau_{mix}\frac{\log T_{max}}{T_{max}}}\right). \tag{62}$$

The proof of Theorem 21 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 9 and 10.

D.2 Corrected Average Reward Tracking Error Analysis

Before we provide the correct version of Theorem 21, we provide the following lemma from Dorfman & Levy (2022) and utilized by Suttle et al. (2023) for Theorem 21 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 22. *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* $\mathbb{E}_{z\sim\mu_{\theta_t},\pi_{\theta_t}}\left[f(\eta, z)\right] = \nabla F(x)$, *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,*

$$\mathbb{E}\left[||f_t^N - \nabla F(\eta_t)||\right] \leq O\left(\sqrt{\log KN}\sqrt{\frac{K}{N}}\right), \tag{63}$$

$$\mathbb{E}\left[||f_t^N - \nabla F(\eta_t)||^2\right] \leq O\left(\log(KN)\frac{K}{N}\right). \tag{64}$$

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

Theorem 23. *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*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\left[(\eta_t - \eta_t^*)^2\right] \leq \mathcal{O}\left(T^{\nu-1}\right) + \mathcal{O}\left(T^{-2(\sigma-\nu)}\right) \tag{65}$$

$$+\widetilde{\mathcal{O}}\left(\tau_{mix}\log T_{max}\right)\mathcal{O}\left(T^{-\nu}\right) \tag{66}$$

$$+\widetilde{\mathcal{O}}\left(\tau_{mix}\frac{\log T_{max}}{T_{max}}\right). \tag{67}$$

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

$$\eta_{t+1} = \eta_t - \gamma_t f_t, \tag{68}$$

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

$$\begin{aligned}(\eta_{t+1} - \eta_{t+1}^*)^2 \leq (1 - 2\gamma_t)(\eta_t - \eta_t^*)^2 + 2\gamma_t(\eta_t - \eta_t^*)(F'(\eta_t) - f_t) + 2(\eta_t - \eta_t^*)(\eta_t^* - \eta_{t+1}^*) \\ + 2(\eta_t^* - \eta_{t+1}^*)^2 + 2(\gamma_t f_t)^2.\end{aligned} \tag{69}$$

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

$$\begin{aligned}\sum_{t=1}^{T} \mathbf{E}[(\eta_t - \eta_t^*)^2] \leq & \underbrace{\sum_{t=1}^{T} \frac{1}{2\gamma_t} \mathbf{E}[(\eta_t - \eta_t^*)^2 - (\eta_t - \eta_t^*)^2]}_{I_1} + \underbrace{\sum_{t=1}^{T} \mathbf{E}[(\eta_t - \eta_t^*)(F'(\eta_t) - f_t)]}_{I_2} \\ & + \underbrace{\sum_{t=1}^{T} \frac{1}{\gamma_t} \mathbf{E}[(\eta_t - \eta_t^*)(\eta_t^* - \eta_{t+1}^*)]}_{I_3} + \underbrace{\sum_{t=1}^{T} \frac{1}{\gamma_t} \mathbf{E}[(\eta_t^* - \eta_{t+1}^*)^2]}_{I_4} + \underbrace{\sum_{t=1}^{T} \gamma_t \mathbf{E}[(f_t)^2]}_{I_5}.\end{aligned} \tag{70}$$

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,

$$I_1 \leq \frac{r_{max}^2}{\gamma_T}, \tag{71}$$

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.

$$|I_2| \leq \sum_{t=1}^{T} \mathbf{E}\left[|(\eta_t - \eta_t^*)| \cdot |(F'(\eta_t) - f_t^{j_{\max}})|\right] \tag{72}$$

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,

$$\begin{aligned}|I_2| &\leq \sum_{t=1}^{T} \mathbf{E}\left[|(\eta_t - \eta_t^*)| \cdot |(F'(\eta_t) - f_t^{j_{\max}})|\right] && (73) \\ &\leq \sum_{t=1}^{T} \mathbf{E}|(\eta_t - \eta_t^*)| \cdot \sum_{t=1}^{T} \mathbf{E}|(F'(\eta_t) - f_t^{j_{\max}})|. && (74) \\ &\leq \left(\sum_{t=1}^{T} \mathbf{E}\left[|(\eta_t - \eta_t^*)|^2\right]\right)^{\frac{1}{2}} \left(\sum_{t=1}^{T} \mathbf{E}\left[|(F'(\eta_t) - f_t^{j_{\max}})|^2\right]\right)^{\frac{1}{2}} && (75)\end{aligned}$$

For $\left(\sum_{t=1}^{T}\mathbb{E}\left[|(F'(\eta_t) - f_t^{j_{\max}})|^2\right]\right)^{\frac{1}{2}}$, we utilize Lemma 22 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:

$$|I_2| \leq \left(\sum_{t=1}^{T}\mathbf{E}\left[|(\eta_t - \eta_t^*)|^2\right]\right)^{\frac{1}{2}} \widetilde{\mathcal{O}}\left(T\tau_{\text{mix}}\frac{\log T_{\max}}{T_{\max}}\right)^{\frac{1}{2}}. \tag{76}$$

Bound on I_3:

$$|I_3| \leq \left(\sum_{t=1}^{T}\mathbf{E}\left[(\eta_t - \eta_t^*)^2\right]\right)^{1/2}\left(L^2G_H^2\sum_{t=1}^{T}\frac{\alpha_t^2}{\gamma_t^2}\right)^{1/2}. \tag{77}$$

Bound on I_4:

$$I_4 \leq L^2G_H^2\sum_{t=1}^{T}\frac{\alpha^2}{\gamma_t}. \tag{78}$$

Bound on I_5:

$$I_5 \leq \sum_{t=1}^{T}\gamma_t\widetilde{\mathcal{O}}\left(R^2\tau_{\text{mix}}^{\theta_t}\log T_{\max}\right). \tag{79}$$

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

$$\sum_{t=1}^{T}\mathbf{E}[(\eta_t - \eta_t^*)^2] \leq 2r_{\max}^2(1+T)^{\nu} + \left[L^2G_H^2 + \widetilde{\mathcal{O}}\left(\tau_{\text{mix}}\log T_{\max}\right)\right]\sum_{t=1}^{T}(1+t)^{-\nu} \tag{80}$$

$$+\left(\sum_{t=1}^{T}\mathbf{E}\left[|(\eta_t - \eta_t^*)|^2\right]\right)^{\frac{1}{2}} \widetilde{\mathcal{O}}\left(T\tau_{\text{mix}}\frac{\log T_{\max}}{T_{\max}}\right)^{\frac{1}{2}} \tag{81}$$

$$+\left(\sum_{t=1}^{T}\mathbf{E}[(\eta_t - \eta_t^*)^2]\right)^{\frac{1}{2}}\left(L^2G_H^2\sum_{t=1}^{T}(1+t)^{-2(\sigma-\nu)}\right)^{\frac{1}{2}}, \tag{82}$$

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

Define

$$Z(T) = \sum_{t=1}^{T}\mathbf{E}[(\eta_t - \eta_t^*)^2], \tag{83}$$

$$F(T) = \frac{L^2G_H^2}{4}\sum_{t=1}^{T}(1+t)^{-2(\sigma-\nu)}, \tag{84}$$

$$G(T) = T\widetilde{\mathcal{O}}\left(\tau_{\text{mix}}\frac{\log T_{\max}}{T_{\max}}\right) \tag{85}$$

$$A(T) = 2r_{\max}^2(1+T)^{\nu} + \left[L^2G_H^2 + \widetilde{\mathcal{O}}\left(\tau_{\text{mix}}\log T_{\max}\right)\right]\sum_{t=1}^{T}(1+t)^{-\nu} \tag{86}$$

$$\tag{87}$$

The inequality can now be written as,

$$Z(T) \leq A(T) + 2\sqrt{Z(T)}\sqrt{F(T)} + 2\sqrt{Z(T)}\sqrt{G(T)} \leq 2A(T) + 16F(T) + 16G(T), \tag{88}$$

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

$$Z(T) \leq 2A(T) + 16F(T) + 16G(T). \tag{89}$$

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,

$$\frac{1}{T}\sum_{t=1}^{T}\mathbf{E}\left[(\eta_t - \eta_t^*)^2\right] \leq \mathcal{O}\left(T^{\nu-1}\right) + \mathcal{O}\left(T^{-2(\sigma-\nu)}\right) + \widetilde{\mathcal{O}}\left(\tau_{mix}\log T_{max}\right)\mathcal{O}\left(T^{-\nu}\right) + \widetilde{\mathcal{O}}\left(\tau_{mix}\frac{\log T_{max}}{T_{max}}\right). \tag{90}$$

□

E 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

$$\theta_{k+1} = \theta_k + \alpha\omega_k \tag{91}$$

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 $\hat{Q}(s, a) - \hat{V}(s)$

F 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) ~ μ_0(·), maximum trajectory length T_max.
for t = 0 to T − 1 do
    Sample level length j_t ~ Geom(1/2)
    for i = 1, . . . , 2^{j_t} do
        Take action a_t^i ~ π_{θ_t}(·|s_t^i)
        Collect next state s_t^{i+1} ~ P(·|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
```