Towards Understanding Asynchronous Advantage Actor-Critic: Convergence and Linear Speedup

Han Shen, Kaiqing Zhang, Mingyi Hong, *Senior Member, IEEE*, and Tianyi Chen

***Abstract*—Asynchronous and parallel implementation of standard reinforcement learning (RL) algorithms is a key enabler of the tremendous success of modern RL. Among many asynchronous RL algorithms, arguably the most popular and effective one is the asynchronous advantage actor-critic (A3C) algorithm. Although A3C is becoming the workhorse of RL, its theoretical properties are still not well-understood, including its non-asymptotic analysis and the performance gain of parallelism (a.k.a. linear speedup). This paper revisits the A3C algorithm and establishes its non-asymptotic convergence guarantees. Under both i.i.d. and Markovian sampling, we establish the local convergence guarantee for A3C in the general policy approximation case and the global convergence guarantee in softmax policy parameterization. Under i.i.d. sampling, A3C obtains sample complexity of $\mathcal{O}(\epsilon^{-2.5}/N)$ per worker to achieve ϵ accuracy, where N is the number of workers. Compared to the best-known sample complexity of $\mathcal{O}(\epsilon^{-2.5})$ for two-timescale AC, A3C achieves *linear speedup*, which justifies the advantage of parallelism and asynchrony in AC algorithms theoretically for the first time. Numerical tests on synthetic environment, OpenAI Gym environments and Atari games have been provided to verify our theoretical analysis.**

***Index Terms*—Reinforcement learning, policy gradient, actor critic, asynchronous parallel method.**

I. Introduction

REINFORCEMENT learning (RL) has achieved impressive performance in many domains such as robotics [30], [33] and video games [32]. However, these empirical successes are often at the expense of significant computation. To unlock high computation capabilities, the state-of-the-art RL approaches rely on sampling data from massive parallel simulators on multiple machines [3], [15], [32], [35]. Empirically, these approaches can significantly *reduce training time* when implemented in an *asynchronous* manner. One popular method that achieves the state-of-art performance is the asynchronous variant of the actor-critic (AC) algorithm, referred to as A3C [32].

Manuscript received 24 February 2022; revised 22 December 2022 and 4 April 2023; accepted 4 April 2023. Date of publication 12 May 2023; date of current version 20 July 2023. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Monica F. Bugallo. This work was supported in part by the Rensselaer-IBM AI Research Collaboration (http://airc.rpi.edu) and in part by IBM AI Horizons Network (http://ibm.biz/AIHorizons). *(Corresponding authors: Tianyi Chen; Han Shen.)*

Han Shen and Tianyi Chen are with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180 USA (e-mail: shenh5@rpi.edu; chentianyi19@gmail.com).

Kaiqing Zhang is with the Laboratory for Information & Decision Systems and Computer Science & Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: kaiqing@csail.mit.edu).

Mingyi Hong is with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: mhong@umn.edu).

Digital Object Identifier 10.1109/TSP.2023.3268475

A3C builds on the original AC algorithm [25]. At a high level, AC simultaneously performs policy optimization (a.k.a. the actor step) using the policy gradient (PG) method [45] and policy evaluation (a.k.a. the critic step) using the temporal difference learning (TD) algorithm [43]. To ensure scalability to large state-action spaces, both actor and critic steps can combine with various function approximation techniques. To ensure stability, AC is often implemented in a two time-scale fashion, where the actor step runs in the slow timescale and the critic step runs in the fast timescale. Similar to other on-policy RL algorithms, AC uses samples generated from the target policy. Thus, data sampling is entangled with the learning procedure, which generates significant *overhead*. To speed up the sampling process of AC, A3C introduces multiple workers with a shared policy, and each worker has its own simulator to perform data sampling. The shared policy can be then updated using samples collected from multiple workers.

Despite the empirical success achieved by A3C, to the best of our knowledge, its theoretical property is not well-understood. The following *theoretical* questions remain unclear: **Q1)** Under what assumption does A3C converge? If so, does it converge to the global optimal solution? **Q2)** What is its convergence rate? **Q3)** Can A3C obtain benefit (or linear speedup) using parallelism and asynchrony?

For **Q3**), we are interested in the *training time linear speedup* with N workers, which is the ratio between the training time using a single worker and that using N workers. Since asynchronous parallelism mitigates the effect of stragglers and keeps workers busy, the training time speedup can be measured roughly by the sample complexity (i.e., computational) linear speedup [29]:

$$\text{Speedup}(N) = \frac{\text{sample complexity with one worker}}{\text{average sample complexity per worker with N workers}}. \tag{1}$$

If $\text{Speedup}(N) = \Theta(N)$, the speedup is linear, and the training time roughly reduces linearly as the number of workers increases. This paper aims to answer this question, towards the goal of providing theoretical justification for the empirical successes of parallel and asynchronous RL.

A. Related Works

The PG method and its global convergence: The global optimality of the stationary points of policy optimization problems has been shown in [7]. Then the finite-time convergence rate for exact PG method with softmax policy was established in [2] by utilizing a gradient-dominance type result under relative entropy regularized objective function. Later, [31] extended this result

TABLE I
COMPARISON OF RESULTS. IN THE TABLE, 'CONSTANT BATCH-SIZE' INDICATES WHETHER OR NOT THE BATCH SIZE IS INDEPENDENT OF THE ACCURACY AND THUS CAN BE A CONSTANT, AND 'SINGLE-LOOP' INDICATES WHETHER OR NOT THE WORK ANALYZES THE SINGLE-LOOP ACTOR-CRITIC METHOD WHERE A SINGLE CRITIC UPDATE IS PERFORMED PER POLICY UPDATE. IN 'SAMPLING METHOD' COLUMN, 'I.I.D.' STANDS FOR I.I.D. SAMPLING FROM THE STATIONARY DISTRIBUTION WHILE 'MARKOVIAN' STANDS FOR SAMPLING FOLLOWING A MARKOV CHAIN

Work	Asynchronous	Single-loop	Constant batch-size	Sampling method	Sample complexity
Z. Yang, 2018 [53]	✗	✗	✗	i.i.d.	Asymptotic
H. Kumar, 2019 [27]	✗	✗	✓	i.i.d.	$\mathcal{O}(\epsilon^{-4})$
S. Qiu, 2019 [36]	✗	✗	✗	Markovian	$\tilde{\mathcal{O}}(\epsilon^{-4})$
Y. Wu, 2020 [49]	✗	✓	✓	Markovian	$\tilde{\mathcal{O}}(\epsilon^{-2.5})$
Theorem 2	✓	✓	✓	i.i.d.	$\mathcal{O}(\epsilon^{-2.5}/N)$
Theorem 4	✓	✓	✓	Markovian	$\tilde{\mathcal{O}}(\epsilon^{-2.5})$

to the entropy regularized setting and established linear convergence rate for exact PG method under softmax parameterization. Later, [8] has proved linear convergence rate for general class PG methods. In the stochastic setting, [55] has established local optimal convergence for stochastic PG with unbiased rollout and increasing step sizes, and [46] has established the global convergence of stochastic neural PG with increasing batch of i.i.d. samples. Later, [54] proved that the minibatch version of PG achieves global convergence with the help of relative entropy regularization. But none of them consider the global convergence of the AC method. On the application side, the PG method has been broadly applied in various settings; see e.g. [12], [13], [14], [19]. In [13], the actor critic method was used to jointly optimize the trajectory, transmission and caching content delivery of the unmanned aerial vehicles. In [14], the policy gradient method was used to jointly optimize the streaming rate and transmission power. In [19], the PG method was used to help the learning of a distribution adaptation strategy. In [12], the PG method is used in a multitask learning algorithm which seeks to improve generalization to new tasks.

Analysis of AC algorithm: AC method was first proposed by [11], [25], with asymptotic convergence guarantees provided in [10], [11], [25]. It was not until recently that the *non-asymptotic* analyses of AC have been established. The finite-sample guarantee for the batch AC algorithm has been established in [22], [27], [53] with i.i.d. sampling. Later, in [36], the finite-sample analysis was established for the double-loop nested AC algorithm under the Markovian setting. An improved analysis for the Markovian setting with minibatch updates has been presented in [51] for the nested AC method. More recently, [49], [52] have provided the first finite-time analyses for the two-timescale AC algorithms under Markov sampling, with both $\tilde{O}(\epsilon^{-2.5})$ sample complexity, which is the best-known sample complexity for two-timescale AC. Through the lens of bi-level optimization, [23] has provided finite-sample guarantees for two-timescale AC, when a *natural* policy gradient step is used in the actor. Recently, [22] also analyzed the single-timescale AC algorithm under an exact critic oracle. On a less relevant line of research, AC-based multi-agent RL has been studied in [16], [37], [56]. However, none of the existing works has analyzed the effect of the asynchronous and parallel updates in AC; see a comparison in Table I.

Parallel and distributed RL methods: In [32], the original A3C method was proposed and became the workhorse in empirical RL. Later, [4] has provided a GPU-version of A3C which significantly decreases training time. Recently, the A3C algorithm is further optimized in modern computers by [41], where a large batch variant of A3C with improved efficiency is also proposed. In [20], an importance weighted distributed AC algorithm IMPALA has been developed to solve a collection of problems with one single set of parameters. A gossip-based distributed AC algorithm has been proposed in [3] which achieves performance competitive to A3C. Additionally, distributed RL is closely related to the multi-agent RL, both of which have a broad range of applications [24], [39], [50]. In [24], a distributed algorithm based on Q-learning was proposed and was shown to achieve convergence under a sparse communication network. In [39], an asynchronous caching approach which utilized PG to find an optimal caching policy was developed. A robust decentralized TD learning method was proposed in [50] to defend against malicious agents in a multi-agent network.

Asynchronous stochastic optimization: For solving general optimization problems, asynchronous stochastic methods have received much attention recently. Due to the possible speedup that can be achieved by asynchronous optimization, it has also been extensively applied to various machine learning areas including RL [32], [39] and distributed learning [48]. The study of asynchronous stochastic methods can be traced back to 1980s [6]. With the batch size M, [1] analyzed asynchronous SGD (async-SGD) for convex functions, and derived a convergence rate of $\mathcal{O}(K^{-\frac{1}{2}}M^{-\frac{1}{2}})$. In [38], a lock-free asynchronous SGD was proven to converge fast under spasity. [21] extended the analysis of [1] to smooth convex with nonsmooth regularization and derived a similar rate. Recent studies by [29] improved upper bound of K_0. In [34], a random parallel algorithm is proposed to solve the problems with large data set size and feature dimension. However, all these works have focused on the single-timescale SGD with a single variable, which cannot capture the stochastic recursion of the AC and A3C algorithms. To best of our knowledge, non-asymptotic analysis of asynchronous two-timescale SGD has remained unaddressed, and its speedup analysis is an uncharted territory.

B. This Work

In this context, we revisit A3C with TD(0) for the critic update. The goal is to provide *non-asymptotic* guarantee and *linear speedup* justification for this popular algorithm.

Our contributions: Compared to the existing literature on both the AC algorithms and the async-SGD, our contributions can be summarized as follows.

c1) We revisit two-timescale A3C and establish its convergence rates with both i.i.d. and Markovian sampling. We first prove the local convergence rate for A3C in the general function approximation case, and then prove that A3C achieves global convergence for the softmax policy parameterization. To the best of our knowledge, this is the first non-asymptotic convergence result for *asynchronous parallel* AC algorithms, and also the first finite time global convergence result for AC.

c2) We characterize the sample complexity of A3C. In the i.i.d. setting, A3C achieves a sample complexity of $\mathcal{O}(\epsilon^{-2.5}/N)$ per worker, where N is the number of workers. Compared to the best-known complexity of $\mathcal{O}(\epsilon^{-2.5})$ for i.i.d. two-timescale AC [23], A3C achieves *linear speedup*, thanks to the parallelism and asynchrony. In the Markovian setting, if delay is bounded, the sample complexity of A3C matches the order of the non-parallel AC algorithm [49].

c3) We test A3C on a synthetic environment to verify our theoretical guarantees with both i.i.d. and Markovian sampling. We also test A3C on the classic control tasks and Atari Games.

Technical challenges: The works [27], [36], [53] analyze the nonparallel nested-loop actor-critic where the critic loop is nested on the actor loop. At each iteration, the critic updates till convergence under a stationary policy. While this work focuses on the A3C algorithm which is a parallel asynchronous single-loop algorithm. For each worker, the actor and critic updates simultaneously at each iteration. Thus compared to [27], [36], [53], this work additionally deals with the policy drift problem for critic update and the asynchrony error. Compared to the recent analysis of nonparallel single-loop AC in [23], [49], [52], several new challenges arise due to parallelism and asynchrony.

Markovian noise coupled with asynchrony and delay: The analysis of two-timescale AC algorithm is non-trivial because of the Markovian noise coupled with both the actor and critic steps. Different from the nonparallel AC that only involves a single Markov chain, A3C introduces multiple Markov chains (one per worker) that mix at different speeds. This is because at a given iteration, workers collect different number of samples and thus their chains mix to different degrees. As we will show later, the worker with the slowest mixing chain will determine the convergence.

Linear speedup for SGD with two coupled sequences: Parallel async-SGD has been shown to achieve linear speedup recently [29], [42]. Different from async-SGD, asynchronous AC is a two-timescale stochastic *semi-gradient* algorithm for solving the more challenging *bilevel* optimization problem (see [23]). The errors induced by asynchrony and delay are intertwined with both the actor and critic updates via a nested structure, which makes the sharp analysis more challenging. Our linear speedup analysis should be also distinguished from that of mini-batch async-SGD [28], where the speedup is a result of *variance reduction* thanks to the larger batch size generated by parallel workers.

Global convergence of A3C under structured problems: We establish global convergence for A3C with softmax policy parameterization and log-barrier regularization. Though the global convergence of policy gradient was established in [31] under the softmax policy without regularization, the result relies on the fact that the policy iterates are uniformly bounded which is true for the deterministic and synchronous algorithm in [31]. While in our case, A3C is a stochastic and asynchronous algorithm and thus we find it difficult to apply the result in [31]. Therefore, we turn to the log-barrier regularization and prove the global convergence result.

II. Preliminaries

A. Markov Decision Process and Policy Gradient

A Markov decision process (MDP) can be described by $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma\}$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}(s'|s,a)$ is the probability of transitioning to $s' \in \mathcal{S}$ from state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$, $r(s,a,s')$ is the reward associated with the transition (s,a,s'), and $\gamma \in [0,1)$ is a discount factor. Throughout the paper, we assume the reward r is upper-bounded by a constant $r_{\max}$. A policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ is defined as a mapping from the state space $\mathcal{S}$ to the probability distribution over the action space $\mathcal{A}$.

Considering discrete time t in an infinite horizon, a policy π can generate a trajectory $(s_0, a_0, s_1, a_1, \ldots)$ with $a_t \sim \pi(\cdot|s_t)$ and $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$. Given a policy π, we define the state and state action value functions as

$$V_\pi(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1}) \mid s_0 = s\right],$$

$$Q_\pi(s,a) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1}) \mid s_0 = s, a_0 = a\right] \quad (2)$$

where $\mathbb{E}$ is taken over the trajectory $(s_0, a_0, s_1, a_1, \ldots)$ generated under policy π. With the above definitions, the advantage function is $A_\pi(s,a) := Q_\pi(s,a) - V_\pi(s)$. With η denoting the initial state distribution, the discounted state visitation measure induced by policy π is defined as $d_\pi(s) := (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s \mid s_0 \sim \eta, \pi)$. We also overload the notation and define the state-action visitation distribution $d_\pi(s,a) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s \mid s_0 \sim \eta, \pi)\pi(a|s)$. In the case where π is parameterized by θ, we use d_θ as shorthand notations for d_{π_θ}.

The goal of RL is to find an optimal policy π^* defined as $\pi^* \in \arg\max_\pi J(\pi) := (1-\gamma)\mathbb{E}_{s\sim\eta}[V_\pi(s)]$, with the optimal return defined as $J^* := \max_\pi J(\pi)$. When the state and action spaces are large, finding the optimal policy π becomes computationally intractable. To overcome the inherent difficulty of learning a function, the policy gradient methods search the best performing policy over a class of parameterized policies. We parameterize the policy with parameter $\theta \in \mathbb{R}^d$, and solve the optimization problem as

$$\max_{\theta \in \mathbb{R}^d} J(\theta) \quad \text{with} \quad J(\theta) := (1-\gamma)\mathbb{E}_{s\sim\eta}[V_{\pi_\theta}(s)]. \quad (3)$$

To maximize $J(\theta)$ with respect to θ, one can update θ using the policy gradient [45]

$$\nabla J(\theta) = \mathbb{E}_{s,a\sim d_\theta}\left[A_{\pi_\theta}(s,a)\psi_\theta(s,a)\right], \quad (4)$$

where $\psi_\theta(s,a) := \nabla \log \pi_\theta(a|s)$. Since computing $\mathbb{E}$ in (4) is expensive if not impossible, popular policy gradient-based algorithms iteratively update θ using stochastic estimate of (4) such as REINFORCE [47] and G(PO)MDP [5].

It is also a common practice to adopt regularization and augment the objective function to

$$J_\lambda(\theta) := J(\theta) - \lambda\mathbb{E}_{s\sim\eta_p}\left[D_{KL}(\pi_p(\cdot|s)|\pi_\theta(\cdot|s))\right] \quad (5)$$

with a regularization constant $\lambda \geq 0$. Here η_p is a prior distribution of states, π_p is a prior policy. The regularization term encourages π_θ to imitate π_p, incorporating prior knowledge into training process. When π_p and η_p are set as uniform distributions, the regularization term is reduced to the relative-entropy regularization widely analyzed in the literature [2], [7], [54]. Moreover, the regularization prevents degenerate solutions that can lead to the pitfall of certain policy parametrization [7]. Given π_p and η_p, we use $R(\theta)$ as a shorthand notation of $-\mathbb{E}_{s\sim\eta_p}[D_{KL}(\pi_p(\cdot|s)|\pi_\theta(\cdot|s))]$.

B. Actor-Critic With Value Function Approximation

Both REINFORCE and G(PO)MDP-based policy gradient algorithms rely on a Monte-Carlo estimate of the value function

$V_{\pi_\theta}(s)$ and thus $\nabla J(\theta)$ by generating a trajectory per iteration. However, policy gradient methods based on Monte-Carlo estimate typically suffer from high variance and large sampling cost. An alternative way is to recursively refine the estimate of $V_{\pi_\theta}(s)$. For a policy π_θ, it is known that $V_{\pi_\theta}(s)$ satisfies the Bellman equation [44], that is

$$V_{\pi_\theta}(s) = \mathbb{E}_{a\sim\pi_\theta(\cdot|s),\, s'\sim\mathcal{P}(\cdot|s,a)}\left[r(s,a,s') + \gamma V_{\pi_\theta}(s')\right]. \quad (6)$$

In practice, when the state space $\mathcal{S}$ is prohibitively large, one cannot afford the computational and memory complexity of computing $V_{\pi_\theta}(s)$ and $A_{\pi_\theta}(s,a)$. To overcome this curse-of-dimensionality, a popular method is to approximate the value function using function approximation techniques. Given the state feature mapping $\phi(\cdot): \mathcal{S} \to \mathbb{R}^{d'}$ for some $d' > 0$, we approximate the value function linearly as $V_{\pi_\theta}(s) \approx \hat{V}_\omega(s) := \phi(s)^\top \omega$, where $\omega \in \mathbb{R}^{d'}$ is the critic parameter.

Given π_θ, the task of finding the best ω such that $V_{\pi_\theta}(s) \approx \hat{V}_\omega(s)$ is usually addressed by TD learning [43]. Given π_θ, the task of finding the best ω such that $V_{\pi_\theta}(s) \approx \hat{V}_\omega(s)$ is usually addressed by TD learning [43]. Formally, we first define

$$A_{\theta,\phi} := \mathbb{E}_{s\sim\mu_{\pi_\theta}, s'\sim\mathcal{P}_{\pi_\theta}}[\phi(s)(\gamma\phi(s') - \phi(s))^\top], \quad (7a)$$

$$b_{\theta,\phi} := \mathbb{E}_{s\sim\mu_{\pi_\theta}, a\sim\pi_\theta}[r(s,a,s')\phi(s)]. \quad (7b)$$

where $\mathcal{P}_{\pi_\theta}(s'|s) := \sum_a \mathcal{P}(s'|s,a)\pi_\theta(a|s)$, and μ_{π_θ} is the stationary distribution of the Markov chain with transition distribution $\mathcal{P}$ and policy π_θ. Then given a policy π_θ, the *exact* TD update takes the following form:

$$\omega_{k+1} = \omega_k + \beta\left(A_{\theta,\phi}\omega_k + b_{\theta,\phi}\right). \quad (8)$$

When analyzing TD, the following standard assumption is often made:

Assumption 1: For all $s \in \mathcal{S}$, the feature vector $\phi(s)$ is normalized so that $\|\phi(s)\|_2 \le 1$. For all eligible θ, the symmetric part of $A_{\theta,\phi}$, denoted as $(A_{\theta,\phi} + A_{\theta,\phi}^\top)/2$, is negative definite and has a largest eigenvalue upper bounded by $-\lambda$.

Assumption 1 is common in analyzing TD with linear function approximation; see e.g., [9], [49], [50]. In fact, as shown in [9], when redundant features are removed such that the feature covariance matrix is full-rank, this assumption is satisfied. With this assumption, $A_{\theta,\phi}$ is full-rank, thus the update in (8) admits a unique stationary point $\omega^*(\theta) = -A_{\theta,\phi}^{-1} b_{\theta,\phi}$. Moreover, there exists a constant $R_\omega := \frac{r_{\max}}{\sqrt{\lambda}(1-\gamma)^{\frac{3}{2}}}$ such that $\|\omega^*(\theta)\|_2 \le R_\omega$.

We often use the stochastic approximation of the TD update in (8). With kth transition defined as $x_k := (s_k, a_k, s_{k+1})$, the corresponding TD target is

$$\hat{\delta}(x_k, \omega_k) := r(s_k, a_k, s_{k+1}) + \gamma\phi(s_{k+1})^\top\omega_k - \phi(s_k)^\top\omega_k \quad (9)$$

and the critic gradient $g(x_k, \omega_k) := \hat{\delta}(x_k, \omega_k)\nabla\hat{V}_{\omega_k}(s_k)$. We update the parameter ω via

$$\omega_{k+1} = \Pi_{R_\omega}\left(\omega_k + \beta g(x_k, \omega_k)\right), \quad (10)$$

where β is the critic step size, and Π_{R_ω} is a projection operator that projects a vector to a l_2 norm ball with radius R_ω. The projection step is often used to control the norm of gradient. In AC, it prevents the actor and critic updates from going too far in the 'wrong' direction; see e.g., [25], [49], [52], [57].

Using the definition that $A_{\pi_\theta}(s,a) = \mathbb{E}_{s'\sim\mathcal{P}}[r(s,a,s') + \gamma V_{\pi_\theta}(s')] - V_{\pi_\theta}(s)$, we can also rewrite (4) as $\nabla J(\theta) = \mathbb{E}_{s,a\sim d_\theta, s'\sim\mathcal{P}}[(r(s,a,s') + \gamma V_{\pi_\theta}(s') - V_{\pi_\theta}(s))\psi_\theta(s,a)]$. Leveraging the value function approximation, we can then approximate the regularized policy gradient as

$$\widehat{\nabla} J_\lambda(\theta) = \widehat{\nabla} J(\theta) + \lambda\widehat{\nabla} R(\theta) = \underbrace{\hat{\delta}(x,\omega)\psi_\theta(s,a)}_{v(x,\theta,\omega)} + \lambda\psi_\theta(x^p). \quad (11)$$

where $v(x,\theta,\omega)$ is an estimator of $\nabla J(\theta)$, and $x^p := (s^p \sim \eta_p, a^p \sim \pi_p(\cdot|s_p))$. Then it is easy to check that $\psi_\theta(x^p)$ is an unbiased estimator of $\nabla R(\theta)$. This gives rise to the policy update

$$\theta_{k+1} = \theta_k + \alpha\left(v(x_k, \theta_k, \omega_k) + \lambda\psi_\theta(x^p)\right), \quad (12)$$

where α is the stepsize for the actor update. To ensure convergence when simultaneously performing critic and actor updates, the stepsizes α and β often decay at two different rates, which is referred to the two-timescale AC [25], [49].

III. A3C Implementation

To speed up the training process, AC can be implemented over N workers in a shared memory setting *without* coordinating among workers [32]. Each worker has its own simulator to perform sampling, and then collaboratively updates the shared policy π_θ using AC updates. As there is no synchronization after each update, the policy used by workers to generate samples may be outdated, which introduces staleness.

Notations on samples: Subscription t in x_t and x_t^p indicates the sample is generated in tth local iteration of a worker. When Markovian sampling is used, subscription t in $x_t = (s_t, a_t, s_{t+1})$ also indicates that it is the tth transition of the local Markov chain. We use k to denote the global counter (or iteration), which increases by one whenever a worker finishes the actor and critic updates in the shared memory. We use subscription (k) in $(s_{(k)}, a_{(k)}, s'_{(k)})$ and $(s^p_{(k)}, a^p_{(k)})$ to indicate the samples used in the kth update.

Algorithm flow: Specifically, we initialize θ_0, ω_0 in the shared memory. Each worker will initialize the simulator with initial state s_0. Without coordination, workers will load θ, ω in the shared memory. The worker then generates samples with either i.i.d. or Markovian sampling method. In Markovian sampling case, we maintain separate Markov chains for actor and critic. For critic, we generate samples following the original transition kernel $\mathcal{P}$. While the actor's chain can be viewed as evolving under a transition kernel $\hat{\mathcal{P}} = \gamma\mathcal{P} + (1-\gamma)\eta$. At each iteration, we have a probability of $1-\gamma$ to reset the chain, thus taking the initial state distribution into account. If the actor's chain evolves under $\mathcal{P}$ like critic, asymptotically the initial distribution η is forgotten, which will introduce an asymptotic error. Once samples are obtained, each worker locally computes the gradients, and then updates the parameters in shared memory asynchronously by

$$\omega_{k+1} = \Pi_{R_\omega}\left(\omega_k + \beta g(x_{(k)}, \omega_{k-\tau_k})\right) \quad (13a)$$

$$\theta_{k+1} = \theta_k + \alpha\left(v(\hat{x}_{(k)}, \theta_{k-\tau_k}, \omega_{k-\tau_k}) + \lambda\psi_{\theta_{k-\tau_k}}(x^p_{(k)})\right) \quad (13b)$$

where τ_k is the delay in the kth actor and critic updates. See A3C in Algorithm 1 and Fig. 1.

Parallel sampling: The AC update (10) and (12) uses samples generated "on-the-fly" from the target policy π_θ, which brings overhead. Compared with (10) and (12), the A3C update (13) allows parallel sampling from N workers, which is the key to linear speedup. We consider the case where only one worker can update parameters in the shared memory at the same time and the update cannot be interrupted. In practice, (13) can also be performed in a mini-batch fashion.

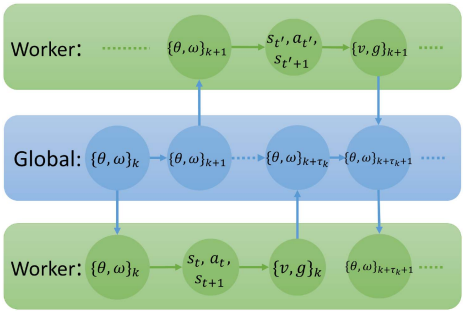

Fig. 1. Implementation of A3C with two workers.

Algorithm 1: A3C: Each Worker's View.

Global initialize: Global counter $k=0$, initial θ_0, ω_0 in the shared memory.
Worker initialize: Counter $t=0$. Sample $s_0\sim\eta$, $\hat{s}_0\sim\eta$.
for $t=0,1,2,\cdots$ **do**
  Read θ,ω in the shared memory.
  option 1 (i.i.d. sampling):
    $x_t=(s_t\sim\mu_{\pi_{\theta_t}}, a_t\sim\pi_{\theta_t}(\cdot|s_t), s'_t\sim\mathcal{P}(\cdot|s_t,a_t))$.
    $\hat{x}_t=(\hat{s}_t\sim d_{\pi_{\theta_t}}, \hat{a}_t\sim\pi_{\theta_t}(\cdot|\hat{s}_t), \hat{s}'_t\sim\mathcal{P}(\cdot|\hat{s}_t,\hat{a}_t))$.
  option 2 (Markovian sampling):
    $x_t=(s_t, a_t\sim\pi_\theta(\cdot|s_t), s_{t+1}\sim\mathcal{P}(\cdot|s_t,a_t))$.
    $\hat{x}_t=(\hat{s}_t, \hat{a}_t\sim\pi_\theta(\cdot|\hat{s}_t), s'_{t+1}\sim\mathcal{P}(\cdot|\hat{s}_t,\hat{a}_t))$.
    With probability γ: $\hat{s}_{t+1}=s'_{t+1}$; Otherwise: $\hat{s}_{t+1}\sim\eta$.
  Compute $g(x_t,\omega)=\hat{\delta}(x_t,\omega)\nabla_\omega\hat{V}_\omega(s_t)$.
  Compute $v(\hat{x}_t,\theta,\omega)=\hat{\delta}(\hat{x}_t,\omega)\psi_\theta(\hat{s}_t,\hat{a}_t)$.
  Compute $\psi_\theta(x_t^p)$ with $x_t^p=(s_t^p\sim\eta_p, a_t^p\sim\pi_p(\cdot|s_t^p))$.
  In the shared memory, perform update (13).
end for

Separate sampling protocols: In Algorithm 1, we maintain separate sampling protocols for actor and critic. This is due to the mismatch between the actor and critic sampling distribution. As indicated by (7), the desired sampling distribution of critic is μ_{π_θ}. The policy gradient (4) requires sampling from d_{π_θ}. However, d_{π_θ} and μ_{π_θ} are in general different, and the difference is non-diminishing. Therefore, if one uses the same samples for actor and critic, either the actor or the critic update will have a non-diminishing bias.

To mitigate the asymptotic bias, it is just natural to choose different sampling protocols for actor and critic. Our theoretical analysis justifies this choice by proving that such sampling method gives unbiased stochastic gradients asymptotically. We also provide experiments to demonstrate the superiority of the separated sampling methods.

IV. Convergence Analysis of A3C

In this section, we analyze the convergence of A3C in both i.i.d. and Markovian settings. Throughout this section, $\mathcal{O}(\cdot)$ contains constants that are independent of N and K_0.

To analyze the performance of A3C, we make the following assumptions.

Assumption 2: There exists K_0 such that the delay at each iteration is bounded by $\tau_k\leq K_0, \forall k$.

Assumption 2 ensures the viability of analyzing the asynchronous update; see the same assumption in e.g., [3], [29], [48]. In practice, the delay usually scales as the number of workers, that is $K_0=\Theta(N)$.

Assumption 3: For any $\theta,\theta'\in\mathbb{R}^d$, $s\in\mathcal{S}$ and $a\in\mathcal{A}$, there exist constants C_ψ, L_ψ, L_π such that: i) $\|\psi_\theta(s,a)\|_2\leq C_\psi$; ii) $\|\psi_\theta(s,a)-\psi_{\theta'}(s,a)\|_2\leq L_\psi\|\theta-\theta'\|_2$; iii) $|\pi_\theta(a|s)-\pi_{\theta'}(a|s)|\leq L_\pi\|\theta-\theta'\|_2$.

Assumption 3 is common in analyzing policy gradient-type algorithms which has also been made by e.g., [2], [55]. This assumption holds for many policy parameterization methods such as tabular softmax policy [2], Gaussian policy [18] and Boltzmann policy [26].

Assumption 4: For any θ, assume the Markov chains with transition kernels $\mathcal{P}$ and $\hat{\mathcal{P}}$ are irreducible and aperiodic under policy π_θ. Then there exist constants $\kappa>0$ and $\rho\in(0,1)$ such that

$$\sup_{s\in\mathcal{S}} d_{TV}\left(\mathbb{P}(s_t\in\cdot|s_0=s,\pi_\theta),\mu_{\pi_\theta}\right)\leq\kappa\rho^t, \tag{14a}$$

and

$$\sup_{s\in\mathcal{S}} d_{TV}\left(\mathbb{P}(\hat{s}_t\in\cdot|\hat{s}_0=s,\pi_\theta),d_{\pi_\theta}\right)\leq\kappa\rho^t. \tag{14b}$$

where s_t is the tth state of the Markov chain with transition kernel $\mathcal{P}$, and $\hat{s}_t$ is the tth state of the Markov chain with transition kernel $\hat{\mathcal{P}}$.

Assumption 4 assumes the Markov chain mixes at a geometric rate. This assumption has also been made by other analysis on Markovian sampling; see e.g. [9], [49]. It is worth noting that the second part of our assumption, that is (14b), holds as long as $\gamma<1$.

We define the critic approximation error as

$$\epsilon_{\text{app}}:=\max_{\theta\in\mathbb{R}^d}\sqrt{\mathbb{E}_{s\sim\mu_\theta}|V_{\pi_\theta}(s)-\hat{V}_{\omega^*_\theta}(s)|^2} \tag{15}$$

where μ_θ is the stationary distribution under π_θ and $\mathcal{P}$. This error captures the quality of the critic function approximation; see also [36], [49], [50]. When the MDP is tabular and the feature matrix is full-rank, the value function V_{π_θ} is in the span of the features. In this case, we have $\epsilon_{\text{app}}=0$.

We first give the convergence result of the critic and actor update under i.i.d. sampling, the proof of which is presented in Section VII.

Theorem 1 (Critic convergence): Suppose Assumptions 1–4 hold. Consider Algorithm 1 with i.i.d. sampling and $\hat{V}_\omega(s)=\phi(s)^\top\omega$. Select step size $\alpha=K^{-\frac{3}{5}}$ and $\beta=K^{-\frac{2}{5}}$. Then it holds that

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left\|\omega_k-\omega^*_{\theta_k}\right\|_2^2=\mathcal{O}\left(\frac{K_0^2}{K^{\frac{4}{5}}}\right)+\mathcal{O}\left(\frac{K_0}{K^{\frac{3}{5}}}\right)+\mathcal{O}\left(\frac{1}{K^{\frac{2}{5}}}\right). \tag{16}$$

Theorem 2 (Actor convergence): Under the same assumptions of Theorem 1, select step size $\alpha=K^{-\frac{3}{5}}$ and $\beta=K^{-\frac{2}{5}}$. Then it holds that

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\|\nabla J_\lambda(\theta_k)\|_2^2 = \mathcal{O}\left(\frac{1}{K^{\frac{2}{5}}}\right)+\mathcal{O}\left(\frac{K_0^2}{K^{\frac{4}{5}}}\right)+\mathcal{O}\left(\frac{K_0}{K^{\frac{3}{5}}}\right)+\mathcal{O}(\epsilon_{\text{app}}). \tag{17}$$

If $K_0=\Theta(N)=\mathcal{O}(K^{\frac{1}{5}})$, then it holds that

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\|\nabla J_\lambda(\theta_k)\|_2^2=\mathcal{O}\left(K^{-\frac{2}{5}}\right)+\mathcal{O}(\epsilon_{\text{app}}) \tag{18}$$

where $\mathcal{O}(\cdot)$ contains constants independent of N and K_0.

Corollary 1 (Linear speedup): To reach ϵ-accuracy in (18), the required number of iterations is $\mathcal{O}(\epsilon^{-2.5})$. Since each iteration of A3C only uses one sample (one transition), the sample complexity is $\mathcal{O}(\epsilon^{-2.5})$, which matches the state-of-the-art sample complexity of two-timescale AC running on one worker. Then under A3C, the average sample complexity per worker is $\mathcal{O}(\epsilon^{-2.5}/N)$ which indicates linear speedup in (1). The negative effect of parameter staleness introduced by parallel asynchrony vanishes asymptotically with the step size. Vanished staleness allows for parallel computing from workers to speedup the training process.

Remark 1 (Comparison to async-SGD analysis): Different from async-SGD (e.g., [29]), the optimal critic parameter ω_θ^* is constantly drifting as θ changes at each iteration. This necessitates setting the actor update to be at a faster time scale than the critic. In this sense, the policy is static relative to the critic asymptotically. In actor update, the gradient $v(x,\theta,\omega)$ is biased because of inexact value function. The bias introduced by the critic optimality gap and the function approximation error correspond to the last two terms in (17).

A. Convergence Result With Markovian Sampling

Due to space limitation, we will directly present the convergence theorem under Markovian sampling and defer the proof to the supplementary material.

Theorem 3 (Critic convergence): Suppose Assumptions 1–4 hold. Consider Algorithm 1 with Markovian sampling and $\hat{V}_\omega(s)=\phi(s)^\top\omega$. Select step size $\alpha=K^{-\frac{3}{5}}$ and $\beta=K^{-\frac{2}{5}}$. Then it holds that

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left\|\omega_k-\omega_k^*\right\|_2^2 = \mathcal{O}\left(\frac{1}{K^{\frac{2}{5}}}\right)+\mathcal{O}\left(\frac{K_0^2\log^2 K}{K^{\frac{3}{5}}}\right)+\mathcal{O}\left(\frac{K_0\log K}{K^{\frac{2}{5}}}\right). \tag{19}$$

The following theorem gives the convergence rate of actor update in Algorithm 1.

Theorem 4 (Actor convergence): Under the same assumptions of Theorem 3, select step size $\alpha=K^{-\frac{3}{5}}$ and $\beta=K^{-\frac{2}{5}}$. Then it holds that

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\|\nabla J_\lambda(\theta_k)\|_2^2 = \mathcal{O}\left(\frac{K_0^2\log^2 K}{K^{\frac{3}{5}}}\right)+\mathcal{O}\left(\frac{K_0\log K}{K^{\frac{2}{5}}}\right)+\mathcal{O}\left(\epsilon_{\text{app}}\right). \tag{20}$$

If we further assume $K_0=\Theta(N)=\mathcal{O}(K^{\frac{1}{5}})$. It holds that

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\|\nabla J_\lambda(\theta_k)\|_2^2=\widetilde{\mathcal{O}}\left(K_0\,K^{-\frac{2}{5}}\right)+\mathcal{O}(\epsilon_{\text{app}}) \tag{21}$$

where $\widetilde{\mathcal{O}}(\cdot)$ hides constants and the logarithmic order of K.

Different from i.i.d. sampling, the stochastic gradients $g(x,\omega)$ and $v(x,\theta,\omega)$ are biased for Markovian sampling, and the bias decreases as the chain mixes. The mixing time corresponds to the logarithmic terms $\log K$ in (19) and (20). Because of asynchrony, at a given iteration, workers collect different number of samples and their chains mix to different degrees. The worker with the slowest mixing chain will determine the rate of convergence. The product of K_0 and $\log K$ in (19) and (20) appears due to the slowest mixing chain. As the last term in (19) dominates other terms asymptotically, the convergence rate degrades as the number of workers increases. While the theoretical linear speedup is difficult to establish in the Markovian setting, we will empirically test it in Section V.

Remark 2 (Challenges compared to AC analysis): Unlike synchronous AC, A3C introduces asynchrony and delay in both the actor and critic updates. At each iteration k, the delayed parameters will introduce extra error in $g(x,\omega_{k-\tau_k})-g(x,\omega_k)$ and $v(x,\theta_{k-\tau_k},\omega_{k-\tau_k})-v(x,\theta_k,\omega_k)$. Furthermore, it also causes delays in sampling since samples are drawn from the delayed policy $\pi_{\theta_{k-\tau_k}}$ instead of π_{θ_k}. This delay will get amplified as every state on the Markov chain is generated by policies with different delays. At local counter t (tth transition on local Markov chain), we compare the chain transition in synchronous and asynchronous settings:

$$\textbf{sync}: s_t \xrightarrow{\theta_t} a_t \xrightarrow{\mathcal{P}} s_{t+1} \xrightarrow{\theta_{t+1}} a_{t+1}\cdots;$$

$$\textbf{async}: s_t \xrightarrow{\theta_{k-\tau_k}} a_t \xrightarrow{\mathcal{P}} s_{t+1} \xrightarrow{\theta_{k+d_t-\tau_{k+d_t}}} a_{t+1}\cdots$$

where k is the global counter at which the local Markov chain takes tth transition, τ_k is the delay of policy used to generate tth local transition, d_t is the number of global updates between two local transitions. Clearly, the parameter delay makes the Markov chain more difficult to analyze.

B. Global Convergence Under Structured Problem

A3C is a gradient ascent type algorithm, thus can only achieve local convergence under a generally non-concave objective function $J_\lambda(\theta)$ w.r.t. θ. However, under some special structured problem, A3C can be shown to achieve global convergence. In this section, we consider the class of MDP which has finite state space and action space. Suppose the policy is parameterized by the softmax function:

$$\pi_\theta(a|s)=\frac{\exp(\theta_{s,a})}{\sum_{s,a}\exp(\theta_{s,a})} \tag{22}$$

where $\theta\in\mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ and $\theta_{s,a}$ is the policy parameter corresponds to pair (s,a). The softmax policy class cannot represent deterministic policies with finite θ. To avoid driving θ to infinity, it is crucial to penalize the deterministic policies with the regularization term introduced in (5). To do so, we set the priors η_p and π_p as uniform distribution on state and action space, then the objective function can be rewritten as

$$J_\lambda(\theta)=J(\theta)+\frac{\lambda}{|\mathcal{S}||\mathcal{A}|}\sum_{s,a}\log\pi_\theta(a|s)+\lambda\log|\mathcal{A}|. \tag{23}$$

Define the state feature matrix $\Phi':=[\phi(s^1),\ \phi(s^2),\ldots,\ \phi(s^{|\mathcal{S}|})]^\top\in\mathbb{R}^{|\mathcal{S}|\times d'}$ of which rows are features. We make the following assumption on Φ'.

Assumption 5: For any eligible θ, there exists $\omega_\theta\in\mathbb{R}^{d'}$ such that $\Phi'\omega_\theta=V_{\pi_\theta}$.

This assumption assumes that the value function V_{π_θ} can be accurately approximated by linear functions. For the assumption to hold, it suffices to select a squared full-rank feature matrix Φ'. It is worth noting that when this assumption does not hold, our result in Theorem 5 holds with an extra error term, which is the function approximation error ϵ_{app}.

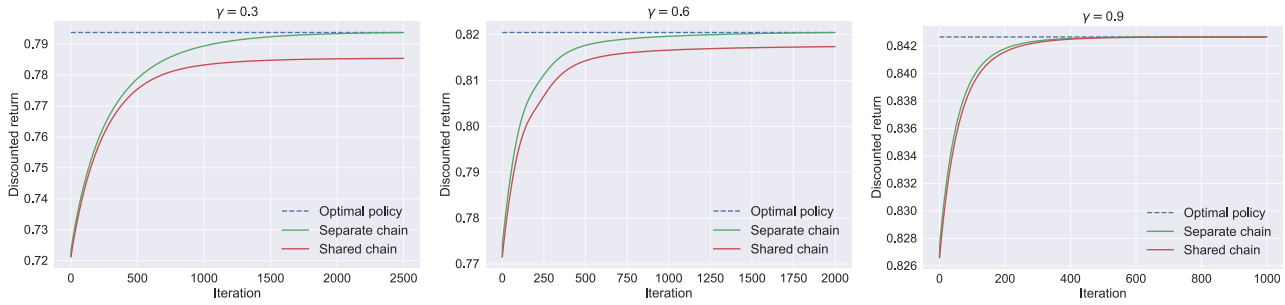

Fig. 2. Algorithm 1 with separate chain sampling (option 2) vs shared chain sampling (setting $\hat{x}_t = x_t$ in the algorithm). The asymptotic error roughly scales propotionally to $1-\gamma$. With a smaller γ, the objective function J becomes more shortsighted, and thus initial state distribution (restarting the chain) plays a more important role. If the actor shares the sample with critic, then a lack of chain restarting will introduce an unavoidable asymptotic error that grows larger as γ becomes smaller. Separate chain sampling works thanks to the random restarting with a probability scaling with γ.

To establish global convergence, a gradient-dominance type condition was proven in [2]:

Lemma 1: With softmax policy parameterization and uniform priors, if $\|\nabla J_\lambda(\theta)\|_2 \leq \frac{\lambda}{2|\mathcal{S}||\mathcal{A}|}$, then $J^* - J(\theta) \leq \epsilon_\lambda := \frac{2\lambda}{1-\gamma}\left\|\frac{d_{\pi^*}}{\eta}\right\|_\infty$.

For an arbitrary accuracy ϵ, if we set $\lambda = \frac{(1-\gamma)\epsilon}{2\|\frac{d_{\pi^*}}{\eta}\|_\infty}$, then we have $\epsilon_\lambda = \epsilon$. Note that in order for $\|\frac{d_{\pi^*}}{\eta}\|_\infty$ to be finite, we need $\eta(s) > 0$ for any $s \in \mathcal{S}$, which can be assumed without loss of generality. In the case where $\eta > 0$ does not hold, one can start with an exploratory initial state distribution $\eta' > 0$ like in [2], and our result still holds. This lemma allows us to establish connection between the gradient norm and optimality gap, giving rise to the following theorem.

Theorem 5: Suppose Assumptions 1, 2 and 4–5 hold. Consider Algorithm 1 with softmax policy and linear critic function $\hat{V}_\omega(s) = \phi(s)^\top \omega$. Select step size $\alpha = K^{-\frac{3}{5}}$, $\beta = K^{-\frac{2}{5}}$ and let $K_0 = \Theta(N) = \mathcal{O}(K^{\frac{1}{5}})$, then it holds

for i.i.d. sampling

$$J^* - \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}\left[J(\theta_k)\right] = \mathcal{O}\left(\lambda^{-2} K^{-\frac{2}{5}}\right) + \epsilon_\lambda, \tag{24a}$$

and for Markovian sampling

$$J^* - \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}\left[J(\theta_k)\right] = \tilde{\mathcal{O}}\left(\lambda^{-2} K_0\, K^{-\frac{2}{5}}\right) + \epsilon_\lambda. \tag{24b}$$

V. Numerical Experiments

We test the impact of separate sampling and the speedup property of A3C in both synthetically generated and Gym environments. The tests on synthetic environment were performed in a 16-core CPU computer, and those on Atari games were run in a 4 GPU computer.

A. Separate Sampling Protocol

We compare the separate chain sampling method in Algorithm 1 with the shared chain sampling method. The shared chain method is simply using the same sample for both actor and critic, i.e., setting $\hat{x}_t = x_t$ in Algorithm 1.

To clearly demonstrate the impact of sampling, we mitigate the impact from other sources such as delay and MDP nonergodicity by considering a synthetic environment with 1 worker.

TABLE II
Hyper-Parameters of A3C in the Atari Games

Hyper-parameters	Value
Number of workers	1,2,4,8,16
Optimizer	Adam
Step size	0.00015
Batch size	20
Discount factor	0.99
Entropy coefficient	0.01
Frame size	80 × 80
Frame skip rate	4
Grayscaling	Yes
Training reward clipping	[-1,1]

In this test, we use the tabular softmax policy parameterization. The synthetic MDP has a state space $|\mathcal{S}| = 10$, an discrete action space of $|\mathcal{A}| = 4$. State features each has a dimension of 10. Elements of the transition matrix, the reward and the state features are randomly sampled from a uniform distribution over (0, 1).

It can be clearly observed from Fig. 2 that using the same sample for actor and critic leads to an asymptotic error scaled with choice of γ. The intuitive explanation is in the caption of Fig. 2. Although when $\gamma \to 1$, the error is small (but still exists), we want our algorithm design to not restrict the choice of γ, and thus adopt the separate chain sampling method.

B. Linear Speedup

Experiment settings: For the synthetic environment, we used linear value function approximation and tabular softmax policy [2]. For CartPole, we used a 3-layer MLP with 128 neurons and sigmoid activation function in each layer. The first two layers are shared for both actor and critic network. For the Atari games, we used a convolution-LSTM network. For network details, see [17].

For the separate sampling protocol test, we have $\alpha = 0.6$ and critic step size $\beta = 0.7$, along with $\lambda = 0.3$. For the speedup tests in synthetic envrionment, we set actor step size $\alpha_k = \frac{0.05}{(1+k)^{0.6}}$ and critic step size $\beta_k = \frac{0.05}{(1+k)^{0.4}}$. In tests of CartPole, we run Algorithm 1 with a minibatch of 20 samples. We update the actor network with a step size of $\alpha_k = \frac{0.01}{(1+k)^{0.6}}$ and critic network with a step size of $\beta_k = \frac{0.01}{(1+k)^{0.4}}$. See Table II for hyper-parameters in Atari game tests.

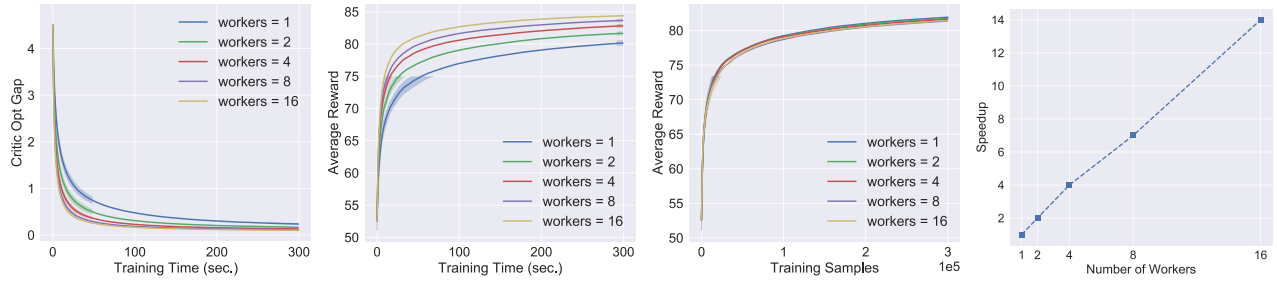

Fig. 3. Convergence results of A3C with i.i.d. sampling in synthetic environment.

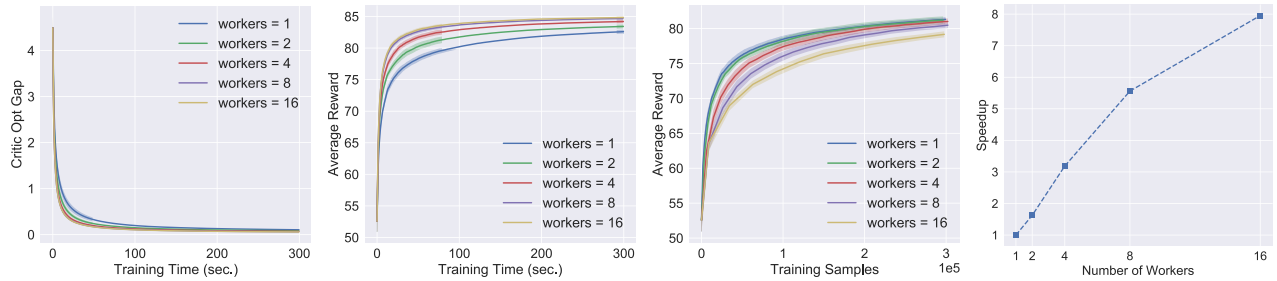

Fig. 4. Convergence results of A3C with Markovian sampling in synthetic environment.

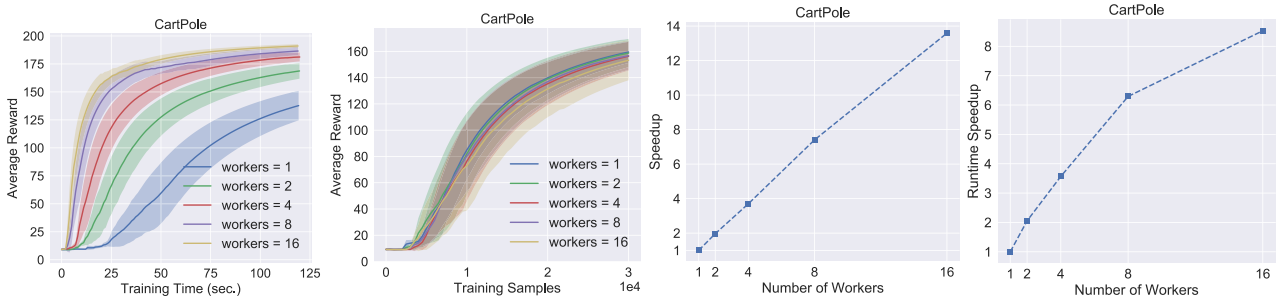

Fig. 5. Speedup of A3C in OpenAI gym classic control task (Carpole).

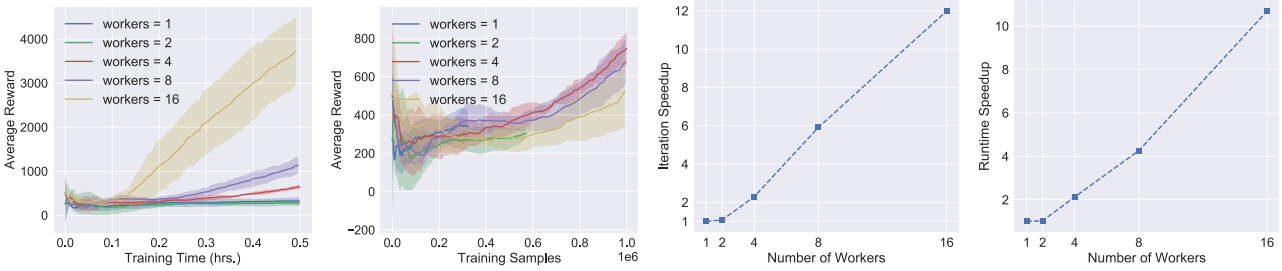

Fig. 6. Speedup of A3C in OpenAI Gym Atari game (Beamrider).

Synthetic environment: We first test the speedup property of A3C in a synthetic environment with $|\mathcal{S}| = 100$, $|\mathcal{A}| = 5$ and state feature with dimension 10. The reward and transition matrix of the MDP are randomly generated in the same way as that in Section V-A. We evaluate the convergence of actor in terms of the average reward and the critic in terms of the gap $\|\omega_k - \omega^*_{\theta_k}\|_2$.

Figs. 3 and 4 show the training time and sample complexity of running A3C with i.i.d. sampling and Markovian sampling respectively. For the speedup plots, we first record the maximum average reward R one worker can achieve in reasonable time. Then we obtain t_n, s_n which are respectively the runtime and samples for n workers to achieve the average reward R. Finally, we calculate the runtime-speedup and speedup for n-workers respectively as t_1/t_n and Ns_1/s_n. All the results are average over 10 Monte-Carlo runs. Fig. 3 shows that the sample complexity of A3C stays the same with different number of workers under i.i.d. sampling. Also, it can be observed from the speedup plot of Fig. 3 that the A3C achieves roughly linear speedup, which is consistent with Corollary 1. The speedup of A3C with Markovian sampling shown in Fig. 4 is roughly linear when number of workers is small.

OpenAI Gym environments: We also test the speedup property of A3C with neural network parametrization in the classic control (Carpole) and the Atari (Breakout and Pong) environments. In Figs. 5–8, each curve was averaged over 5 Monte-Carlo runs with 95% confidence interval. Figs. 5–8 show

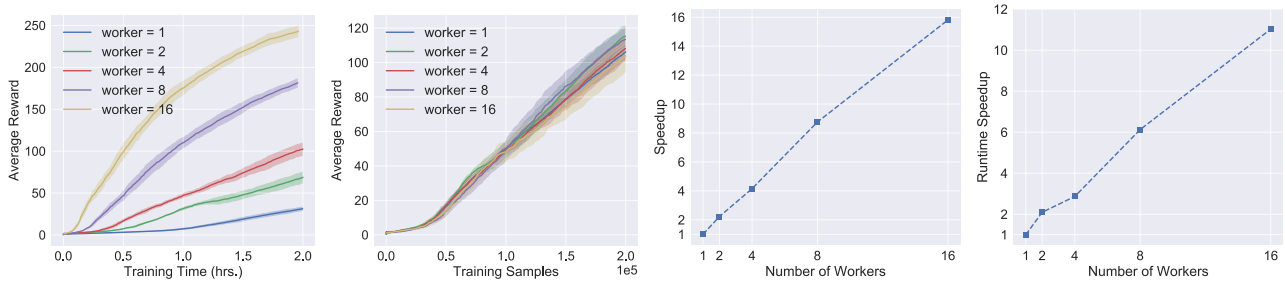

Fig. 7. Speedup of A3C in OpenAI Gym Atari game (Breakout).

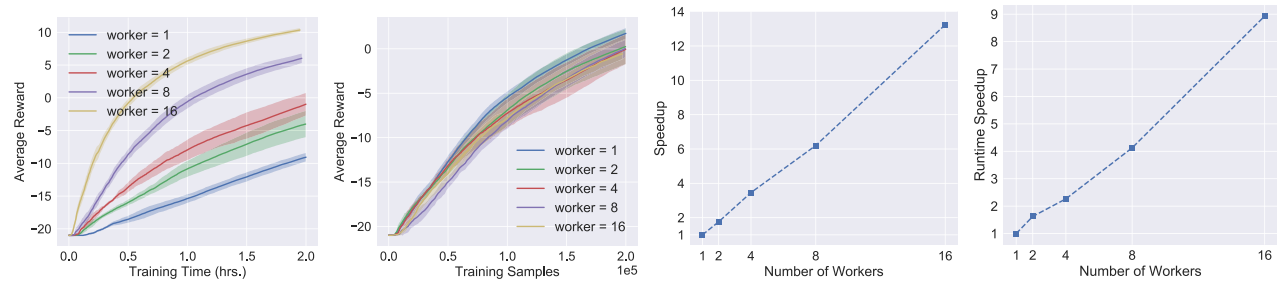

Fig. 8. Speedup of A3C in OpenAI Gym Atari game (Pong).

the speedup of A3C under different number of workers, where the average reward is computed by taking the running average of test rewards. It can be observed from the figures that the speedup is sometimes sub-linear. Our theorem suggests this is due to the Markovian sampling error that scales with the number of workers. If according to [38], another reason might be related to the sparsity of the gradient. It has been observed in [38] that linear speedup is easier to achieve with a sparse gradient. While in our applications, the rewards are not sparse and we use a dense neural network which may not give a sparse gradient. Other reasons are related to hardware limits, see e.g., [29].

VI. Conclusion

This paper revisits the A3C algorithm. With linear value function approximation, the convergence of the A3C algorithm has been established under both i.i.d. and Markovian sampling settings. Under i.i.d. sampling, A3C achieves linear speedup compared to the best-known sample complexity of AC, theoretically justifying the benefit of parallelism and asynchrony for the first time. Under Markov sampling, such a linear speedup can be observed in most benchmark tasks.

One limitation of this paper is that theoretical linear speedup cannot be established in the Markovian setting. This motivates two interesting directions: i) developing new tools of analyzing two-timescale SGD with Markov sampling; and, ii) designing better algorithms than A3C to achieve better speedup.

VII. Proof

In this section, we provide the convergence analysis of A3C under i.i.d. sampling (Theorems 1 and 2) and the global convergence of A3C (Theorem 5). We defer the proof of Theorem 3 and Theorem 4 to the supplementary material.

A. Preliminary Lemmas

We first give a proposition regarding the L_λ-Lipschitz continuity of the regularized policy gradient under proper assumptions, which has been shown by [2], [55].

Proposition 1: Suppose Assumption 3 hold. For any $\theta, \theta' \in \mathbb{R}^d$, we have $\|\nabla J_\lambda(\theta) - \nabla J_\lambda(\theta')\|_2 \leq L_\lambda \|\theta - \theta'\|_2$, where L_λ is a positive constant.

We then directly give a proposition that will be useful in the main proof, the justification of which is deferred to the supplementary material.

Proposition 2: Suppose Assumption 1, 3 and 4 hold. For any $\theta_1, \theta_2 \in \mathbb{R}^d$, we have

$$\|\omega^*_{\theta_1} - \omega^*_{\theta_2}\|_2 \leq L_\omega \|\theta_1 - \theta_2\|_2,$$

where $L_\omega := 2r_{\max}|\mathcal{A}|L_\pi(\lambda^{-1} + \lambda^{-2}(1+\gamma))(1 + \log_\rho \kappa^{-1} + (1-\rho)^{-1})$.

B. Proof of Theorem 1

As compared to the works in [27], [36], [53] that analyze the nested-loop AC, this proof analyzes the asynchronous single-loop AC and additionally deals with asynchrony error along with the policy drift problem in the critic update.

We first define the exact TD update as:

$$\overline{g}(x, \omega) := \mathbb{E}_{s\sim\mu_\theta, a\sim\pi_\theta, s'\sim\mathcal{P}} \left[g(x, \omega) \right]. \tag{25}$$

The critic update in Algorithm 1 can be written as:

$$\omega_{k+1} = \Pi_{R_\omega} \left(\omega_k + \beta g(x_{(k)}, \omega_{k-\tau_k}) \right), \tag{26}$$

where τ_k is the delay of the parameters used in evaluating the kth stochastic gradient, and $x_{(k)} := (s_{(k)}, a_{(k)}, s'_{(k)})$ is the sample used to evaluate the stochastic gradient at kth update.

Proof: Using ω^*_k as shorthand notation of $\omega^*_{\theta_k}$, we start with the optimality gap

$$\|\omega_{k+1} - \omega^*_{k+1}\|_2^2$$
$$= \|\Pi_{R_\omega} \left(\omega_k + \beta g(x_{(k)}, \omega_{k-\tau_k}) \right) - \omega^*_{k+1}\|_2^2$$

$$\leq \|\omega_k + \beta g(x_{(k)}, \omega_{k-\tau_k}) - \omega^*_{k+1}\|_2^2$$
$$= \|\omega_k - \omega^*_k + \beta g(x_{(k)}, \omega_{k-\tau_k}) + \omega^*_k - \omega^*_{k+1}\|_2^2$$

Expanding RHS of the last equality gives

$$\begin{aligned}
&\|\omega_{k+1} - \omega^*_{k+1}\|_2^2 \\
&= \|\omega_k - \omega^*_k\|_2^2 + \left\|\omega^*_k - \omega^*_{k+1} + \beta g(x_{(k)}, \omega_{k-\tau_k})\right\|_2^2 \\
&\quad +2\left\langle \omega_k - \omega^*_k, \omega^*_k - \omega^*_{k+1}\right\rangle + 2\beta\left\langle \omega_k - \omega^*_k, g(x_{(k)}, \omega_{k-\tau_k})\right\rangle \\
&= \|\omega_k - \omega^*_k\|_2^2 + 2\beta\left\langle \omega_k - \omega^*_k, g(x_{(k)}, \omega_{k-\tau_k})\right\rangle \\
&\quad +2\left\langle \omega_k - \omega^*_k, \omega^*_k - \omega^*_{k+1}\right\rangle + 2\left\|\omega^*_k - \omega^*_{k+1}\right\|_2^2 + 2C_\delta^2\beta^2. \quad (27)
\end{aligned}$$

The second term in (27) can be decomposed as

$$\begin{aligned}
&\left\langle \omega_k - \omega^*_k, g(x_{(k)}, \omega_{k-\tau_k})\right\rangle \\
&= \left\langle \omega_k - \omega^*_k, \overline{g}(\theta_k, \omega_k)\right\rangle + \left\langle \omega_k - \omega^*_k, g(x_{(k)}, \omega_k) - \overline{g}(\theta_k, \omega_k)\right\rangle \\
&\quad + \left\langle \omega_k - \omega^*_k, g(x_{(k)}, \omega_{k-\tau_k}) - g(x_{(k)}, \omega_k)\right\rangle. \quad (28)
\end{aligned}$$

We first bound $\langle \omega_k - \omega^*_k, \overline{g}(\theta_k, \omega_k)\rangle$ in (28) as

$$\begin{aligned}
&\left\langle \omega_k - \omega^*_k, \overline{g}(\theta_k, \omega_k)\right\rangle \\
&\quad = \left\langle \omega_k - \omega^*_k, \overline{g}(\theta_k, \omega_k) - \overline{g}(\theta_k, \omega^*_k)\right\rangle
\end{aligned}$$

where the equality is due to $\overline{g}(\theta, \omega^*_\theta) = A_{\theta,\phi}\omega^*_\theta + b = 0$. By definition of $\overline{g}$, we can continue to write

$$\begin{aligned}
&\left\langle \omega_k - \omega^*_k, \overline{g}(\theta_k, \omega_k)\right\rangle \\
&\quad = \left\langle \omega_k - \omega^*_k, \mathbb{E}\left[\phi(s)\left(\gamma\phi(s') - \phi(s)\right)^\top\right](\omega_k - \omega^*_k)\right\rangle \\
&\quad = \left\langle \omega_k - \omega^*_k, A_{\pi_{\theta_k}}(\omega_k - \omega^*_k)\right\rangle \\
&\quad \leq -\lambda\|\omega_k - \omega^*_k\|_2^2, \quad (29)
\end{aligned}$$

where the last inequality follows Assumption 1.

We then bound the third term in (28) as

$$\begin{aligned}
&\left\langle \omega_k - \omega^*_k, g(x_{(k)}, \omega_{k-\tau_k}) - g(x_{(k)}, \omega_k)\right\rangle \\
&= \left\langle \omega_k - \omega^*_k, \left(\gamma\phi(s'_{(k)}) - \phi(s_{(k)})\right)^\top(\omega_{k-\tau_k} - \omega_k)\phi(s_{(k)})\right\rangle \\
&\leq (1+\gamma)\|\omega_k - \omega^*_k\|_2\|\omega_{k-\tau_k} - \omega_k\|_2 \\
&\leq (1+\gamma)\|\omega_k - \omega^*_k\|_2 \sum_{i=k-\tau_k}^{k-1}\beta\|g(x_i, \omega_{i-\tau_i})\|_2 \\
&\leq 2C_\delta K_0\beta\|\omega_k - \omega^*_k\|_2, \quad (30)
\end{aligned}$$

where constant $C_\delta := r_{\max} + (1+\gamma)\max\{\frac{r_{\max}}{1-\gamma}, R_\omega\}$, then the last inequality follows from

$$\begin{aligned}
\|g(x,\omega)\|_2 &\leq |r(x) + \gamma\phi(s')^\top\omega - \phi(s)^\top\omega| \\
&\leq r_{\max} + (1+\gamma)R_\omega \leq C_\delta \quad (31)
\end{aligned}$$

and likewise, we have $\|\overline{g}(x,\omega)\|_2 \leq C_\delta$.

Substituting (30) and (29) into (28) gives

$$\begin{aligned}
&\left\langle \omega_k - \omega^*_k, g(x_{(k)}, \omega_{k-\tau_k})\right\rangle \\
&\leq -\lambda\|\omega_k - \omega^*_k\|_2^2 + 2C_\delta K_0\beta\|\omega_k - \omega^*_k\|_2 \\
&\quad + \left\langle \omega_k - \omega^*_k, g(x_{(k)}, \omega_k) - \overline{g}(\theta_k, \omega_k)\right\rangle. \quad (32)
\end{aligned}$$

Next we jointly bound the third and fourth term in (27) as

$$\begin{aligned}
&\left\langle \omega_k - \omega^*_k, \omega^*_k - \omega^*_{k+1}\right\rangle + \left\|\omega^*_k - \omega^*_{k+1}\right\|_2^2 \\
&\leq \|\omega_k - \omega^*_k\|_2\left\|\omega^*_k - \omega^*_{k+1}\right\|_2 + \left\|\omega^*_k - \omega^*_{k+1}\right\|_2^2 \\
&\leq 2L_\omega\|\omega_k - \omega^*_k\|_2\|\theta_k - \theta_{k+1}\|_2 + 2L_\omega^2\|\theta_k - \theta_{k+1}\|_2^2 \\
&\leq 2L_\omega C_p\alpha\|\omega_k - \omega^*_k\|_2 + 2L_\omega^2 C_p^2\alpha^2, \quad (33)
\end{aligned}$$

where constant $C_p := C_\delta C_\psi + \lambda C_\psi$. The second inequality is due to the L_ω-Lipschitz continuity of ω^*_θ shown in Proposition 2 in the supplementary, and the last inequality follows the fact that

$$\begin{aligned}
\|\theta_k - \theta_{k+1}\|_2 &= \alpha\|v(\hat{x}_{(k)}, \theta_{k-\tau_k}, \omega_{k-\tau_k}) + \lambda\psi_{\theta_{k-\tau_k}}(x^p_{(k)})\|_2 \\
&\leq \alpha C_p. \quad (34)
\end{aligned}$$

Substituting (32) and (33) into (27), and taking expectation on both sides yield

$$\begin{aligned}
&\mathbb{E}\|\omega_{k+1} - \omega^*_{k+1}\|_2^2 \\
&\leq (1 - 2\lambda\beta)\mathbb{E}\|\omega_k - \omega^*_k\|_2^2 \\
&\quad + 2\beta\left(C_1\frac{\alpha}{\beta} + C_2K_0\beta\right)\mathbb{E}\|\omega_k - \omega^*_k\|_2 \\
&\quad + 2\beta\mathbb{E}\left\langle \omega_k - \omega^*_k, g(x_{(k)}, \omega_k) - \overline{g}(\theta_k, \omega_k)\right\rangle + C_q\beta^2, \quad (35)
\end{aligned}$$

where $C_1 := L_\omega C_p$, $C_2 := 2C_\delta$ and $C_q := 2C_\delta^2 + 2L_\omega^2 C_p^2\frac{\alpha^2}{\beta^2}$.

For brevity, we use $x \sim \theta$ to denote $s \sim \mu_\theta$, $a \sim \pi_\theta$ and $s' \sim \mathcal{P}$ in this proof. Consider the third term in (35) conditioned on $\theta_k, \omega_k, \theta_{k-\tau_k}$. We bound it as

$$\begin{aligned}
&\mathbb{E}\left[\left\langle \omega_k - \omega^*_k, g(x_{(k)}, \omega_k) - \overline{g}(\theta_k, \omega_k)\right\rangle|\theta_k, \omega_k, \theta_{k-\tau_k}\right] \\
&= \left\langle \omega_k - \omega^*_k, \mathbb{E}_{x_{(k)}\sim\theta_{k-\tau_k}}\left[g(x_{(k)}, \omega_k)|\omega_k\right] - \overline{g}(\theta_k, \omega_k)\right\rangle \\
&= \left\langle \omega_k - \omega^*_k, \overline{g}(\theta_{k-\tau_k}, \omega_k) - \overline{g}(\theta_k, \omega_k)\right\rangle \\
&\leq \|\omega_k - \omega^*_k\|_2\|\overline{g}(\theta_{k-\tau_k}, \omega_k) - \overline{g}(\theta_k, \omega_k)\|_2 \\
&\leq 2R_\omega\left\|\mathbb{E}_{x\sim\theta_{k-\tau_k}}[g(x, \omega_k)] - \mathbb{E}_{x\sim\theta_k}[g(x, \omega_k)]\right\|_2 \\
&\leq 2R_\omega\sup_x\|g(x, \omega_k)\|_2\left\|\mu_{\theta_{k-\tau_k}}\otimes\pi_{\theta_{k-\tau_k}} - \mu_{\theta_k}\otimes\pi_{\theta_k}\right\|_{TV} \\
&\leq 4R_\omega C_\delta d_{TV}(\mu_{\theta_{k-\tau_k}}\otimes\pi_{\theta_{k-\tau_k}}, \mu_{\theta_k}\otimes\pi_{\theta_k}), \quad (36)
\end{aligned}$$

where second last inequality follows the definition of TV norm.

Define constant $C_3 := 2R_\omega C_\delta|\mathcal{A}|L_\pi(1 + \log_\rho\kappa^{-1} + (1-\rho)^{-1})$. Then by following the third item in [49, Lemma A.1], we can write (36) as

$$\begin{aligned}
&\mathbb{E}\left[\left\langle \omega_k - \omega^*_k, g(x_{(k)}, \omega_k) - \overline{g}(\theta_k, \omega_k)\right\rangle|\theta_k, \omega_k, \theta_{k-\tau_k}\right] \\
&\leq 4R_\omega C_\delta d_{TV}(\mu_{\theta_{k-\tau_k}}\otimes\pi_{\theta_{k-\tau_k}}\otimes\mathcal{P}, \mu_{\theta_k}\otimes\pi_{\theta_k}\otimes\mathcal{P}) \\
&\leq C_3\|\theta_{k-\tau_k} - \theta_k\|_2 \\
&\leq C_3\sum_{i=k-\tau_k}^{k-1}\alpha\|g(x_i, \omega_{i-\tau_i})\|_2 \\
&\leq C_3C_\delta K_0\alpha, \quad (37)
\end{aligned}$$

Taking total expectation on both sides of (37) and substituting it into (35) yield

$$\begin{aligned}
\mathbb{E}\|\omega_{k+1} - \omega^*_{k+1}\|_2^2 &\leq (1 - 2\lambda\beta)\mathbb{E}\|\omega_k - \omega^*_k\|_2^2 \\
&\quad + 2\beta\left(C_1\frac{\alpha}{\beta} + C_2K_0\beta\right)\mathbb{E}\|\omega_k - \omega^*_k\|_2 \\
&\quad + 2C_3C_\delta K_0\beta\alpha + C_q\beta^2. \quad (38)
\end{aligned}$$

which along with the fact $\alpha = \frac{1}{(K+1)^{\frac{3}{5}}}$ and $\beta = \frac{1}{(K+1)^{\frac{2}{5}}}$ implies

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left\|\omega_k-\omega_k^*\right\|_2^2=\mathcal{O}\left(\frac{K_0^2}{K^{\frac{4}{5}}}\right)+\mathcal{O}\left(\frac{K_0}{K^{\frac{3}{5}}}\right)+\mathcal{O}\left(\frac{1}{K^{\frac{2}{5}}}\right). \tag{39}$$

This completes the proof. □

C. Proof of Theorem 2

As compared to the works in [27], [36], [53] that analyze the nonparallel AC, this proof analyzes the parallel asynchronous AC and additionally deals with asynchrony error in the actor update and establishes the linear speedup.

We first define the 'optimal' TD target as:

$$\delta(x,\theta) := r(s,a,s') + \gamma V_{\pi_\theta}(s') - V_{\pi_\theta}(s).$$

The update in Algorithm 1 can be written as:

$$\theta_{k+1} = \theta_k + \alpha\hat{\delta}(\hat{x}_{(k)},\omega_{k-\tau_k})\psi_{\theta_{k-\tau_k}}(\hat{s}_{(k)},\hat{a}_{(k)}) + \alpha\lambda\psi_{\theta_{k-\tau_k}}(x^p_{(k)}). \tag{40}$$

For brevity, we use ω_k^* as shorthand notation of $\omega_{\theta_k}^*$ in this proof. We also write the delayed score function $\psi_{\theta_{k-\tau_k}}(\hat{s}_{(k)},\hat{a}_{(k)})$ as $\psi_{k-\tau_k}$. Then we are ready to give the convergence proof.

Proof: From L_λ-Lipschitz continuity of regularized policy gradient shown in Proposition 1, we have:

$$J_\lambda(\theta_{k+1}) - J_\lambda(\theta_k) \geq \langle\nabla J_\lambda(\theta_k),\theta_{k+1}-\theta_k\rangle - \frac{L_\lambda}{2}\|\theta_{k+1}-\theta_k\|_2^2.$$

By the update rule (40), we can rewrite the first term in RHS of the above inequality as

$$\begin{aligned}
&J_\lambda(\theta_{k+1}) - J_\lambda(\theta_k)\\
&\geq \alpha\big\langle\nabla J_\lambda(\theta_k),\big(\hat{\delta}(\hat{x}_{(k)},\omega_{k-\tau_k})-\hat{\delta}(\hat{x}_{(k)},\omega_k^*)\big)\psi_{k-\tau_k}\big\rangle\\
&\quad+\alpha\big\langle\nabla J_\lambda(\theta_k),\hat{\delta}(\hat{x}_{(k)},\omega_k^*)\psi_{k-\tau_k}+\lambda\psi_{\theta_{k-\tau_k}}(x^p_{(k)})\big\rangle\\
&\quad-\frac{L_\lambda}{2}\|\theta_{k+1}-\theta_k\|_2^2\\
&\geq \alpha\big\langle\nabla J_\lambda(\theta_k),\big(\hat{\delta}(\hat{x}_{(k)},\omega_{k-\tau_k})-\hat{\delta}(\hat{x}_{(k)},\omega_k^*)\big)\psi_{k-\tau_k}\big\rangle\\
&\quad+\alpha\big\langle\nabla J_\lambda(\theta_k),\hat{\delta}(\hat{x}_{(k)},\omega_k^*)\psi_{k-\tau_k}+\lambda\psi_{\theta_{k-\tau_k}}(x^p_{(k)})\big\rangle\\
&\quad-\frac{L_\lambda}{2}C_p^2\alpha^2,
\end{aligned}$$

where the last inequality follows the definition of C_p in (34).

Taking expectation on both sides of the last inequality yields

$$\begin{aligned}
&\mathbb{E}[J_\lambda(\theta_{k+1})] - \mathbb{E}[J_\lambda(\theta_k)]\\
&\geq \alpha\underbrace{\mathbb{E}\big\langle\nabla J_\lambda(\theta_k),\big(\hat{\delta}(\hat{x}_{(k)},\omega_{k-\tau_k})-\hat{\delta}(\hat{x}_{(k)},\omega_k^*)\big)\psi_{k-\tau_k}\big\rangle}_{I_1}\\
&\quad+\alpha\underbrace{\mathbb{E}\big\langle\nabla J_\lambda(\theta_k),\hat{\delta}(\hat{x}_{(k)},\omega_k^*)\psi_{k-\tau_k}+\lambda\nabla R(\theta_{k-\tau_k})\big\rangle}_{I_2}\\
&\quad-\frac{L_\lambda}{2}C_p^2\alpha^2.
\end{aligned}\tag{41}$$

where we used the fact that $\mathbb{E}[\psi_{\theta_{k-\tau_k}}(x^p_{(k)})|\theta_{k-\tau_K}] = \nabla R(\theta_{k-\tau_k})$.

We first decompose I_1 as

$$\begin{aligned}
&\mathbb{E}\big\langle\nabla J_\lambda(\theta_k),\big(\hat{\delta}(\hat{x}_{(k)},\omega_{k-\tau_k})-\hat{\delta}(\hat{x}_{(k)},\omega_k^*)\big)\psi_{k-\tau_k}\big\rangle\\
&=\underbrace{\mathbb{E}\big\langle\nabla J_\lambda(\theta_k),\big(\hat{\delta}(\hat{x}_{(k)},\omega_{k-\tau_k})-\hat{\delta}(\hat{x}_{(k)},\omega_k)\big)\psi_{k-\tau_k}\big\rangle}_{I_1^{(1)}}\\
&\quad+\underbrace{\mathbb{E}\big\langle\nabla J_\lambda(\theta_k),\big(\hat{\delta}(\hat{x}_{(k)},\omega_k)-\hat{\delta}(\hat{x}_{(k)},\omega_k^*)\big)\psi_{k-\tau_k}\big\rangle}_{I_1^{(2)}}.
\end{aligned}\tag{42}$$

We bound $I_1^{(1)}$ as

$$\begin{aligned}
I_1^{(1)} &= \mathbb{E}\big\langle\nabla J_\lambda(\theta_k),\big(\gamma\phi(\hat{s}'_{(k)})-\phi(\hat{s}_{(k)})\big)^\top\\
&\quad(\omega_{k-\tau_k}-\omega_k)\psi_{k-\tau_k}\big\rangle\\
&\geq -2C_\psi\mathbb{E}\left[\|\nabla J_\lambda(\theta_k)\|_2\|\omega_k-\omega_{k-\tau_k}\|_2\right]\\
&\geq -2C_\psi C_\delta K_0\beta\mathbb{E}\|\nabla J_\lambda(\theta_k)\|_2.
\end{aligned}\tag{43}$$

The last inequality follows

$$\begin{aligned}
\|\omega_k-\omega_{k-\tau_k}\|_2 &= \left\|\sum_{i=k-\tau_k}^{k-1}(\omega_{i+1}-\omega_i)\right\|_2\\
&\leq \sum_{i=k-\tau_k}^{k-1}\|\beta g(x_i,\omega_{i-\tau_i})\|_2\\
&\leq \beta K_0 C_\delta,
\end{aligned}\tag{44}$$

where the last inequality is due to (31).

Similarly, we can bound $I_1^{(2)}$ as

$$I_1^{(2)} \geq -(1+\gamma)C_\psi\mathbb{E}\left[\|\nabla J_\lambda(\theta_k)\|_2\|\omega_k-\omega_k^*\|_2\right]. \tag{45}$$

Collecting lower bounds of $I_1^{(1)}$ and $I_1^{(2)}$ gives

$$\begin{aligned}
I_1 &\geq -2C_\psi\mathbb{E}\left[\|\nabla J_\lambda(\theta_k)\|_2\left(C_\delta K_0\beta+\|\omega_k-\omega_k^*\|_2\right)\right]\\
&\geq -\frac{1}{2}\mathbb{E}\|\nabla J_\lambda(\theta_k)\|_2^2 - 2C_\psi^2\mathbb{E}\left[\left(C_\delta K_0\beta+\|\omega_k-\omega_k^*\|_2\right)^2\right]\\
&\geq -\frac{1}{2}\mathbb{E}\|\nabla J_\lambda(\theta_k)\|_2^2 - 4C_\psi^2C_\delta^2K_0^2\beta^2 - 4C_\psi^2\mathbb{E}\|\omega_k-\omega_k^*\|_2^2,
\end{aligned}\tag{46}$$

where the the second and third inequality follow Young's inequality.

Now we consider I_2. We first decompose I_2 as

$$\begin{aligned}
&\mathbb{E}\big\langle\nabla J_\lambda(\theta_k),\hat{\delta}(\hat{x}_{(k)},\omega_k^*)\psi_{k-\tau_k}+\lambda\nabla R(\theta_{k-\tau_k})\big\rangle\\
&=\underbrace{\mathbb{E}\big\langle\nabla J_\lambda(\theta_k),\big(\hat{\delta}(\hat{x}_{(k)},\omega_k^*)-\hat{\delta}(\hat{x}_{(k)},\omega_{k-\tau_k}^*)\big)\psi_{k-\tau_k}\big\rangle}_{I_2^{(1)}}\\
&\quad+\underbrace{\mathbb{E}\big\langle\nabla J_\lambda(\theta_k),\big(\hat{\delta}(\hat{x}_{(k)},\omega_{k-\tau_k}^*)-\delta(\hat{x}_{(k)},\theta_{k-\tau_k})\big)\psi_{k-\tau_k}\big\rangle}_{I_2^{(2)}}\\
&\quad+\underbrace{\mathbb{E}\big\langle\nabla J_\lambda(\theta_k),\delta(\hat{x}_{(k)},\theta_{k-\tau_k})\psi_{k-\tau_k}+\lambda\nabla R(\theta_{k-\tau_k})-\nabla J_\lambda(\theta_k)\big\rangle}_{I_2^{(3)}}\\
&\quad+\|\nabla J_\lambda(\theta_k)\|_2^2.
\end{aligned}\tag{47}$$

By the definition of $\hat{\delta}$ in (9), we can write $I_2^{(1)}$ as

$$I_2^{(1)} = \mathbb{E}\big\langle\nabla J_\lambda(\theta_k),\big(\gamma\phi(\hat{s}'_{(k)})-\phi(\hat{s}_{(k)})\big)^\top \times\big(\omega_k^*-\omega_{k-\tau_k}^*\big)\psi_{k-\tau_k}\big\rangle$$

$$\geq -L_V C_\psi(1+\gamma)\mathbb{E}\left\|\omega_k^* - \omega_{k-\tau_k}^*\right\|_2$$
$$\geq -L_V L_\omega C_\psi(1+\gamma)\mathbb{E}\|\theta_k - \theta_{k-\tau_k}\|_2$$
$$\geq -L_V L_\omega C_\psi C_p(1+\gamma)K_0\alpha, \tag{48}$$

where $L_V := \frac{r_{\max}}{1-\gamma}C_\psi + \lambda C_\psi$ is the trivial upper bound of $\|\nabla J_\lambda(\theta)\|_2$ and $\|\nabla J(\theta)\|_2$. The second last inequality follows from Proposition 2 and the last inequality uses (34) as

$$\|\theta_k - \theta_{k-\tau_k}\|_2 \leq \sum_{i=k-\tau_k}^{k-1}\|\theta_{i+1}-\theta_i\|_2$$
$$= \sum_{i=k-\tau_k}^{k-1}\alpha\|\hat{\delta}(\hat{x}_i,\omega_{i-\tau_i})\psi_{\theta_{i-\tau_i}}(\hat{s}_i,\hat{a}_i)\|_2$$
$$\leq \sum_{i=k-\tau_k}^{k-1}\alpha C_p \leq C_p K_0\alpha. \tag{49}$$

We bound $I_2^{(2)}$ as

$$I_2^{(2)} = \mathbb{E}\left\langle \nabla J_\lambda(\theta_k), \left(\hat{\delta}(\hat{x}_{(k)},\omega_{k-\tau_k}^*) - \delta(\hat{x}_{(k)},\theta_{k-\tau_k})\right)\psi_{k-\tau_k}\right\rangle$$
$$\geq -L_V C_\psi \mathbb{E}\left|\hat{\delta}(\hat{x}_{(k)},\omega_{k-\tau_k}^*) - \delta(\hat{x}_{(k)},\theta_{k-\tau_k})\right|$$
$$\geq -L_V C_\psi\Big(\gamma\mathbb{E}\left|\phi(\hat{s}'_{(k)})^\top\omega_{k-\tau_k}^* - V_{\pi_{\theta_{k-\tau_k}}}(\hat{s}'_{(k)})\right|$$
$$+\mathbb{E}\left|V_{\pi_{\theta_{k-\tau_k}}}(\hat{s}_{(k)}) - \phi(\hat{s}_{(k)})^\top\omega_{k-\tau_k}^*\right|\Big)$$
$$\geq -L_V C_\psi\left(\gamma\sqrt{\mathbb{E}\left|\phi(\hat{s}'_{(k)})^\top\omega_{k-\tau_k}^* - V_{\pi_{\theta_{k-\tau_k}}}(\hat{s}'_{(k)})\right|^2}\right.$$
$$\left.+\sqrt{\mathbb{E}\left|V_{\pi_{\theta_{k-\tau_k}}}(\hat{s}_{(k)}) - \phi(\hat{s}_{(k)})^\top\omega_{k-\tau_k}^*\right|^2}\right)$$
$$\geq -L_V C_\psi(1+\gamma)\epsilon_{\text{app}}. \tag{50}$$

Using the fact that

$$\mathbb{E}\left[\delta(\hat{x}_{(k)},\theta_{k-\tau_k})\psi_{k-\tau_k}\middle|\theta_{k-\tau_k},\theta_k\right]$$
$$= \mathbb{E}_{\,d_{\theta_{k-\tau_k}}}\left[A_{\pi_{\theta_{k-\tau_k}}}(\hat{s}_{(k)},\hat{a}_{(k)})\psi_{k-\tau_k}\middle|\theta_{k-\tau_k},\theta_k\right]$$
$$= \nabla J(\theta_{k-\tau_k}), \tag{51}$$

then we can write

$$I_2^{(3)} = \left\langle\nabla J_\lambda(\theta_k), \nabla J_\lambda(\theta_{k-\tau_k}) - \nabla J_\lambda(\theta_k)\right\rangle$$
$$\geq -\|\nabla J_\lambda(\theta_k)\|_2\|\nabla J_\lambda(\theta_{k-\tau_k}) - \nabla J_\lambda(\theta_k)\|_2$$
$$\geq -L_V L_\lambda\|\theta_{k-\tau_k} - \theta_k\|_2 \geq -L_V L_\lambda C_p K_0\alpha, \tag{52}$$

where the second last inequality is due to L_λ-Lipschitz continuity of policy gradient shown in Proposition 1, and the last inequality follows (49).

Collecting lower bounds of $I_2^{(1)}$, $I_2^{(2)}$ and $I_2^{(3)}$ gives

$$I_2 \geq -D_1 K_0\alpha - L_V C_\psi(1+\gamma)\epsilon_{\text{app}}, \tag{53}$$

where constant $D_1 := L_V L_\omega C_\psi C_p(1+\gamma) + L_V L_\lambda C_p$.

Substituting (46) and (53) into (41) yields

$$\mathbb{E}[J_\lambda(\theta_{k+1})] - \mathbb{E}[J_\lambda(\theta_k)]$$
$$\geq \frac{\alpha}{2}\mathbb{E}\|\nabla J_\lambda(\theta_k)\|_2^2 - 4C_\psi^2\alpha\mathbb{E}\|\omega_k - \omega_k^*\|_2^2 - 4C_\psi^2 C_\delta^2 K_0^2\alpha\beta^2$$
$$-2L_V C_\psi\epsilon_{\text{app}}\alpha - \left(D_1 K_0 + \frac{L_\lambda}{2}C_p^2\right)\alpha^2. \tag{54}$$

After telescoping, we have

$$\sum_{k=1}^K\frac{1}{2}\mathbb{E}\|\nabla J_\lambda(\theta_k)\|_2^2 \leq \frac{1}{\alpha}\left(J^* - J_\lambda(\theta_{K_0})\right)$$
$$+4C_\psi^2\sum_{k=1}^K\mathbb{E}\|\omega_k - \omega_k^*\|_2^2 + 4\,K C_\psi^2 C_\delta^2 K_0^2\beta^2$$
$$+2\,K L_V C_\psi\epsilon_{\text{app}} + K\left(D_1 K_0 + \frac{L_\lambda}{2}C_p^2\right)\alpha. \tag{55}$$

Select $\alpha = K^{-\frac{3}{5}}$ and $\beta = K^{-\frac{2}{5}}$, we have

$$\frac{1}{K}\sum_{k=1}^K\mathbb{E}\|\nabla J_\lambda(\theta_k)\|_2^2 = \mathcal{O}\left(\frac{1}{K^{\frac{2}{5}}}\right) + \mathcal{O}\left(\frac{1}{K}\sum_{k=1}^K\mathbb{E}\|\omega_k - \omega_k^*\|_2^2\right)$$
$$+\mathcal{O}\left(\frac{K_0^2}{K^{\frac{4}{5}}}\right) + \mathcal{O}\left(\frac{K_0}{K^{\frac{3}{5}}}\right) + \mathcal{O}(\epsilon_{\text{app}}). \tag{56}$$

This completes the proof. □

D. Proof of Theorem 5

Proof: We define an event E_k as $\|\nabla J_\lambda(\theta_k)\| \leq \frac{\lambda}{2|\mathcal{S}||\mathcal{A}|}$ and its complement E_k^c as $\|\nabla J_\lambda(\theta_k)\| > \frac{\lambda}{2|\mathcal{S}||\mathcal{A}|}$. We use $\mathbf{1}_{E_k}$ to indicate whether the event happens or not, i.e. $\mathbf{1}_{E_k} = 1$ if E_k happens and $\mathbf{1}_{E_k} = 0$ if E_k^c happens. Then we have

$$\sum_{k=1}^K\mathbb{E}\left[J^* - J(\theta_k)\right]$$
$$= \sum_{k=1}^K\mathbb{E}\left[\left(J^* - J(\theta_k)\right)\mathbf{1}_{E_k}\right] + \sum_{k=1}^K\mathbb{E}\left[\left(J^* - J(\theta_k)\right)\mathbf{1}_{E_k^c}\right]$$
$$\leq \frac{2\lambda}{1-\gamma}\left\|\frac{d_{\pi^*}}{\eta}\right\|_\infty\sum_{k=1}^K\mathbb{E}\left[\mathbf{1}_{E_k}\right] + \sum_{k=1}^K\mathbb{E}\left[\left(J^* - J(\theta_k)\right)\mathbf{1}_{E_k^c}\right]$$
$$\leq \frac{2\lambda}{1-\gamma}\left\|\frac{d_{\pi^*}}{\eta}\right\|_\infty\sum_{k=1}^K\mathbb{E}\left[\mathbf{1}_{E_k}\right] + J^*\sum_{k=1}^K\mathbb{E}\left[\mathbf{1}_{E_k^c}\right]$$
$$\leq \frac{2\lambda}{1-\gamma}\left\|\frac{d_{\pi^*}}{\eta}\right\|_\infty K + J^*\sum_{k=1}^K\mathbb{E}\left[\mathbf{1}_{E_k^c}\right], \tag{57}$$

where the first inequality follows from Lemma 1.

Now it suffices to bound $\sum_{k=1}^K\mathbb{E}[\mathbf{1}_{E_k^c}]$.

$$\sum_{k=1}^K\mathbb{E}\|\nabla J_\lambda(\theta_k)\|^2 \geq \sum_{k=1}^K\mathbb{E}\left[\|\nabla J_\lambda(\theta_k)\|^2\mathbf{1}_{E_k^c}\right]$$
$$\geq \sum_{k=1}^K\frac{\lambda^2}{4|\mathcal{S}|^2|\mathcal{A}|^2}\mathbb{E}\left[\mathbf{1}_{E_k^c}\right] \tag{58}$$

Substituting the above inequality into (57) and dividing both sides by K give

$$\frac{1}{K}\sum_{k=1}^K\mathbb{E}\left[J^* - J(\theta_k)\right]$$
$$\leq \frac{2\lambda}{1-\gamma}\left\|\frac{d_{\pi^*}}{\eta}\right\|_\infty + \frac{4|\mathcal{S}|^2|\mathcal{A}|^2}{\lambda^2}\frac{1}{K}\sum_{k=1}^K\mathbb{E}\|\nabla J_\lambda(\theta_k)\|^2. \tag{59}$$

It is known that the softmax policy satisfies Assumption 3, thus we immediately know that Theorems 2 and 4 hold. Furthermore,

by assumption 5, we have $\phi(s)^\top \omega_\theta^* = V_{\pi_\theta}(s)$ and thus $\epsilon_{\text{app}} = 0$. Applying Theorem 2 and 4 to (59) completes the proof. □

APPENDIX A
PRELIMINARY LEMMAS

A. Geometric Mixing

The operation $p \otimes q$ denotes the product between two distributions $p(x)$ and $q(y)$, i.e. $(p \otimes q)(x, y) = p(x) \cdot q(y)$.

Lemma 2: Suppose Assumption 4 holds. For any $\theta \in \mathbb{R}^d$, we have

$$\sup_{s_0 \in \mathcal{S}} d_{TV}\left(\mathbb{P}((s_t, a_t, s_{t+1}) \in \cdot | s_0, \pi_\theta), \mu_\theta \otimes \pi_\theta \otimes \mathcal{P}\right) \leq \kappa \rho^t. \tag{60a}$$

and

$$\sup_{s_0 \in \mathcal{S}} d_{TV}\left(\mathbb{P}((\hat{s}_t, \hat{a}_t, s'_{t+1}) \in \cdot | s_0, \pi_\theta), d_\theta \otimes \pi_\theta \otimes \mathcal{P}\right) \leq \kappa \rho^t. \tag{60b}$$

where (s_t, a_t, s_{t+1}) is the tth transition on the Makov chain with transition kernel $\mathcal{P}$. $(\hat{s}_t, \hat{a}_t)$ is the tth state-action pair on Markov chain with transition kernel $\hat{\mathcal{P}}$, and $s'_{t+1} \sim \mathcal{P}(\cdot | \hat{s}_t, \hat{a}_t)$.

Proof: We start with

$$\begin{aligned}
&\sup_{s_0 \in \mathcal{S}} d_{TV}\left(\mathbb{P}((s_t, a_t, s_{t+1}) = \cdot | s_0, \pi_\theta), \mu_\theta \otimes \pi_\theta \otimes \mathcal{P}\right) \\
&= \sup_{s_0 \in \mathcal{S}} d_{TV}\left(\mathbb{P}(s_t = \cdot | s_0, \pi_\theta) \otimes \pi_\theta \otimes \mathcal{P}, \mu_\theta \otimes \pi_\theta \otimes \mathcal{P}\right) \\
&= \sup_{s_0 \in \mathcal{S}} \frac{1}{2} \int_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \int_{s' \in \mathcal{S}} \Big| \mathbb{P}(s_t = ds | s_0, \pi_\theta) \pi_\theta(a|s) \mathcal{P}(ds'|s, a) \\
&\quad - \mu_\theta(ds) \pi_\theta(a|s) \mathcal{P}(ds'|s, a) \Big| \\
&= \sup_{s_0 \in \mathcal{S}} d_{TV}\left(\mathbb{P}(s_t \in \cdot | s_0, \pi_\theta), \mu_\theta\right) \\
&\leq \kappa \rho^t,
\end{aligned} \tag{61}$$

Inequality (60a) along with the fact that the stationary distribution of the Markov chain with transition probability $\hat{\mathcal{P}}$ and policy π_θ is simply d_θ immediately implies (60b). This completes the proof. □

For the use in the later proof, given $K > 0$, we first define m_K as:

$$m_K := \min\left\{m \in \mathbb{N}^+ | \kappa \rho^{m-1} \leq \min\{\alpha, \beta\}\right\}, \tag{62}$$

where κ and ρ are constants defined in (4). m_K is the minimum number of samples needed for the Markov chain to approach the stationary distribution so that the bias incurred by the Markovian sampling is small enough.

B. Lipschitz Continuity of Critic

We provide a justification for Lipschitz continuity of ω_θ^* in the following proposition.

Proposition 3 (Restatement of Proposition 2): Suppose Assumption 1, 3 and 4 hold. For any $\theta_1, \theta_2 \in \mathbb{R}^d$, we have

$$\|\omega_{\theta_1}^* - \omega_{\theta_2}^*\|_2 \leq L_\omega \|\theta_1 - \theta_2\|_2,$$

where $L_\omega := 2r_{\max}|\mathcal{A}|L_\pi(\lambda^{-1} + \lambda^{-2}(1+\gamma))(1 + \log_\rho \kappa^{-1} + (1-\rho)^{-1})$.

Proof: We use A_1, A_2, b_1 and b_2 as shorthand notations of $A_{\pi_{\theta_1}}$, $A_{\pi_{\theta_2}}$, $b_{\pi_{\theta_1}}$ and $b_{\pi_{\theta_2}}$ respectively. By Assumption 1, $A_{\theta,\phi}$ is invertible for any $\theta \in \mathbb{R}^d$, so we can write $\omega_\theta^* = -A_{\theta,\phi}^{-1} b_{\theta,\phi}$. Then we have

$$\begin{aligned}
&\|\omega_1^* - \omega_2^*\|_2 \\
&= \| - A_1^{-1} b_1 + A_2^{-1} b_2\|_2 \\
&= \| - A_1^{-1} b_1 - A_1^{-1} b_2 + A_1^{-1} b_2 + A_2^{-1} b_2\|_2 \\
&= \| - A_1^{-1}(b_1 - b_2) - (A_1^{-1} - A_2^{-1}) b_2\|_2 \\
&\leq \|A_1^{-1}(b_1 - b_2)\|_2 + \|(A_1^{-1} - A_2^{-1}) b_2\|_2 \\
&\leq \|A_1^{-1}\|_2 \|b_1 - b_2\|_2 + \|A_1^{-1} - A_2^{-1}\|_2 \|b_2\|_2 \\
&= \|A_1^{-1}\|_2 \|b_1 - b_2\|_2 + \|A_1^{-1}(A_2 - A_1) A_2^{-1}\|_2 \|b_2\|_2 \\
&\leq \|A_1^{-1}\|_2 \|b_1 - b_2\|_2 + \|A_1^{-1}\|_2 \|A_2^{-1}\|_2 \|b_2\|_2 \|(A_2 - A_1)\|_2 \\
&\leq \lambda^{-1} \|b_1 - b_2\|_2 + \lambda^{-2} r_{\max} \|A_1 - A_2\|_2,
\end{aligned} \tag{63}$$

where the last inequality follows Assumption 1, and the fact that

$$\begin{aligned}
\|b_2\|_2 &= \|\mathbb{E}[r(s, a, s') \phi(s)]\|_2 \leq \mathbb{E} \|r(s, a, s') \phi(s)\|_2 \\
&\leq \mathbb{E}\left[|r(s, a, s')| \|\phi(s)\|_2\right] \leq r_{\max}.
\end{aligned}$$

Denote (s^1, a^1, s'^1) and (s^2, a^2, s'^2) as samples drawn with θ_1 and θ_2 respectively, i.e. $s^1 \sim \mu_{\theta_1}$, $a^1 \sim \pi_{\theta_1}$, $s'^1 \sim \mathcal{P}$ and $s^2 \sim \mu_{\theta_2}$, $a^2 \sim \pi_{\theta_2}$, $s'^2 \sim \mathcal{P}$. Then we have

$$\begin{aligned}
&\|b_1 - b_2\|_2 \\
&= \left\|\mathbb{E}\left[r(s^1, a^1, s'^1)\phi(s^1)\right] - \mathbb{E}\left[r(s^2, a^2, s'^2)\phi(s^2)\right]\right\|_2 \\
&\leq r_{\max} \|\mathbb{P}((s^1, a^1, s'^1) \in \cdot) - \mathbb{P}((s^2, a^2, s'^2) \in \cdot)\|_{TV} \\
&= 2 r_{\max} d_{TV}\left(\mu_{\theta_1} \otimes \pi_{\theta_1} \otimes \mathcal{P}, \mu_{\theta_2} \otimes \pi_{\theta_2} \otimes \mathcal{P}\right) \\
&\leq 2 r_{\max} |\mathcal{A}| L_\pi (1 + \log_\rho \kappa^{-1} + (1-\rho)^{-1}) \|\theta_1 - \theta_2\|_2,
\end{aligned} \tag{64}$$

where the first inequality follows the definition of total variation (TV) norm, and the last inequality follows Lemma A.1. in [49]. Similarly we have:

$$\begin{aligned}
&\|A_1 - A_2\|_2 \\
&\leq 2(1+\gamma) d_{TV}\left(\mu_{\theta_1} \otimes \pi_{\theta_1}, \mu_{\theta_2} \otimes \pi_{\theta_2}\right) \\
&= (1+\gamma) |\mathcal{A}| L_\pi (1 + \log_\rho \kappa^{-1} + (1-\rho)^{-1}) \|\theta_1 - \theta_2\|_2.
\end{aligned} \tag{65}$$

Substituting (64) and (65) into (63) completes the proof. □

APPENDIX B
PROOF OF RESULTS UNDER MARKOV SAMPLING

In this section, we prove the convergence of A3C under Markovian sampling. As compared to previous work on non-parallel AC [27], [36], [49], [53], the proof for Thoerem 3 and Theorem 4 additionally deal with the asynchrony error coupled with Markovian noise. As compared to the nested-loop AC analysis [27], [36], [53], this proof also addionally deals with the policy drift problem in critic update.

A. Proof of Theorem 3

Given the definition in Section VII-B, we now give the convergence proof of critic update in Algorithm 1 with linear function approximation and Markovian sampling.

The following Lemma will be used in the proof of Theorem 3. Due to space limitation, we directly present the result and defer the proof to the supplementary material of the online version of this paper [40].

Lemma 3: For any $m \geq 1$ and $k \geq (K_0+1)m + K_0 + 1$, we have

$$\begin{aligned}&\mathbb{E}\left\langle \omega_k - \omega^*_{\theta_k}, g(x_{(k)}, \omega_k) - \overline{g}(\theta_k, \omega_k)\right\rangle \\ &\leq C_4 \mathbb{E}\|\theta_k - \theta_{k-d_m}\|_2 + C_5 \sum_{i=\tau_k}^{d_m} \mathbb{E}\|\theta_{k-i} - \theta_{k-d_m}\|_2 \\ &\quad + C_6 \mathbb{E}\|\omega_k - \omega_{k-d_m}\|_2 + C_7 \kappa \rho^{m-1},\end{aligned}$$

where constant $d_m \leq (K_0+1)m + K_0$, and $C_4 := 2C_\delta L_\omega + 4R_\omega C_\delta |\mathcal{A}| L_\pi \quad (1 + \log_\rho \kappa^{-1} \quad + (1-\rho)^{-1})$, $C_5 := 4R_\omega C_\delta |\mathcal{A}| L_\pi$ and $C_6 := 4(1+\gamma)\, R_\omega + 2C_\delta$, $C_7 := 8R_\omega C_\delta$.

Now we start to prove Theorem 3.

Proof: By following the derivation of (35), we have

$$\begin{aligned}\mathbb{E}\|\omega_{k+1} - \omega^*_{k+1}\|_2^2 \leq &(1-2\lambda\beta)\mathbb{E}\,\|\omega_k - \omega^*_k\|_2^2 \\ &+ 2\beta\left(C_1\frac{\alpha}{\beta} + C_2 K_0 \beta\right)\mathbb{E}\,\|\omega_k - \omega^*_k\|_2 \\ &+ 2\beta \mathbb{E}\left\langle \omega_k - \omega^*_k, g(x_{(k)}, \omega_k) - \overline{g}(\theta_k, \omega_k)\right\rangle + C_q \beta^2, \quad (66)\end{aligned}$$

where $C_1 := C_p L_\omega$, $C_2 := C_\delta(1+\gamma)$ and $C_q := 2C_\delta^2 + 2L_\omega^2 C_p^2 \max_{(k)} \frac{\alpha^2}{\beta^2}$.

Now we consider the third item in the last inequality. For some $m \in \mathbb{N}^+$, we define $M := (K_0+1)m + K_0$. Following Lemma 3, for some $d_m \leq M$ and positive constants C_4, C_5, C_6, C_7, we have

$$\begin{aligned}&\mathbb{E}\left\langle \omega_k - \omega^*_k, g(x_{(k)}, \omega_k) - \overline{g}(\theta_k, \omega_k)\right\rangle \\ &\leq C_4 \mathbb{E}\|\theta_k - \theta_{k-d_m}\|_2 + C_5 \sum_{i=\tau_k}^{d_m} \mathbb{E}\|\theta_{k-i} - \theta_{k-d_m}\|_2 \\ &\quad + C_6 \mathbb{E}\|\omega_k - \omega_{k-d_m}\|_2 + C_7 \kappa \rho^{m-1} \\ &\leq C_4 \sum_{i=k-d_m}^{k-1} \mathbb{E}\|\theta_{i+1} - \theta_i\|_2 + C_5 \sum_{i=\tau_k}^{d_m-1} \sum_{j=k-d_m}^{k-i-1} \mathbb{E}\|\theta_{j+1} - \theta_j\|_2 \\ &\quad + C_6 \sum_{i=k-d_m}^{k-1} \mathbb{E}\|\omega_{i+1} - \omega_i\|_2 + C_7 \kappa \rho^{m-1} \\ &\leq C_4 d_m C_p \alpha + C_5 (d_m - \tau_k)^2 C_p \alpha + C_6 d_m C_\delta \beta + C_7 \kappa \rho^{m-1} \\ &\leq \left(C_4\, M + C_5\, M^2\right) C_p \alpha + C_6\, M C_\delta \beta + C_7 \kappa \rho^{m-1}, \quad (67)\end{aligned}$$

where the last inequality is due to $\tau_k \geq 0$ and $d_m \leq M$.

Further letting $m = m_K$ which is defined in (62) yields

$$\begin{aligned}&\mathbb{E}\left\langle \omega_k - \omega^*_k, g(x_{(k)}, \omega_k) - \overline{g}(\theta_k, \omega_k)\right\rangle \\ &= \left(C_4 M_K + C_5 M_K^2\right) C_p \alpha + C_6 C_\delta M_K \beta + C_7 \kappa \rho^{m_K - 1} \\ &\leq \left(C_4 M_K + C_5 M_K^2\right) C_p \alpha + C_6 C_\delta M_K \beta + C_7 \alpha, \quad (68)\end{aligned}$$

where $M_K = (K_0+1)m_K + K_0$, and the last inequality follows the from $m_K = \mathcal{O}(\log K)$.

Substituting (68) into (66) gives

$$\begin{aligned}&\mathbb{E}\|\omega_{k+1} - \omega^*_{k+1}\|_2^2 \quad (69)\\ &\leq (1-2\lambda\beta)\mathbb{E}\,\|\omega_k - \omega^*_k\|_2^2 \\ &\quad + 2\beta\left(C_1\frac{\alpha}{\beta} + C_2 K_0 \beta\right)\mathbb{E}\,\|\omega_k - \omega^*_k\|_2 \\ &\quad + 2\beta\left(\left(C_4 M_K + C_5 M_K^2\right) C_p \alpha + C_6 C_\delta M_K \beta + C_7 \alpha\right) + C_q \beta^2. \quad (70)\end{aligned}$$

Select $\alpha = K^{-\frac{3}{5}}$ and $\beta = K^{-\frac{2}{5}}$. After telescoping, we have

$$\begin{aligned}&\frac{1}{K}\sum_{k=1}^{K} \mathbb{E}\,\|\omega_k - \omega^*_k\|_2^2 \\ &= \mathcal{O}\left(\frac{1}{K^{\frac{2}{5}}}\right) + \mathcal{O}\left(\frac{K_0^2 \log^2 K}{K^{\frac{3}{5}}}\right) + \mathcal{O}\left(\frac{K_0 \log K}{K^{\frac{2}{5}}}\right).\end{aligned}$$

This completes the proof. □

B. Proof of Theorem 4

Given the definition in Section VII-C, we now give the convergence proof of actor update in Algorithm 1 with linear value function approximation and Markovian sampling method.

The following lemmas will be used in the proof of Theorem 4. Due to space limitation, we directly present the results and defer the proof to the supplementary material of [40].

Lemma 4: For any $m \geq 1$ and $k \geq (K_0+1)m + K_0 + 1$, we have

$$\begin{aligned}&\mathbb{E}\left\langle \nabla J_\lambda(\theta_k), \left(\hat{\delta}(\hat{x}_{(k)}, \omega^*_k) - \delta(\hat{x}_{(k)}, \theta_k)\right)\psi_{\theta_{k-\tau_k}}(\hat{s}_{(k)}, \hat{a}_{(k)})\right\rangle \\ &\geq -D_2 \mathbb{E}\|\theta_{k-\tau_k} - \theta_{k-d_m}\|_2 - D_3 \mathbb{E}\|\theta_k - \theta_{k-d_m}\|_2 \\ &\quad - D_4 \sum_{i=\tau_k}^{d_m} \mathbb{E}\|\theta_{k-i} - \theta_{k-d_m}\|_2 - D_5 \kappa \rho^{m-1} - L_V C_\psi (1+\gamma)\epsilon_{\text{app}},\end{aligned}$$

where $D_2 := 2L_V L_\psi C_\delta$, $D_3 := (2C_\delta C_\psi L_\lambda + L_V C_\psi (L_\omega + L_V)(1+\gamma))$, $D_4 := 2L_V C_\psi C_\delta |\mathcal{A}| L_\pi$ and $D_5 := 4L_V C_\psi C_\delta$.

Lemma 5: For any $m \geq 1$ and $k \geq (K_0+1)m + K_0 + 1$, we have

$$\begin{aligned}&\mathbb{E}\left\langle \nabla J_\lambda(\theta_k), \delta(\hat{x}_{(k)}, \theta_k)\psi_{\theta_{k-\tau_k}}(\hat{s}_{(k)}, \hat{a}_{(k)}) - \nabla J(\theta_k)\right\rangle \\ &\geq -D_6 \mathbb{E}\|\theta_{k-\tau_k} - \theta_{k-d_m}\|_2 - D_7 \mathbb{E}\|\theta_k - \theta_{k-d_m}\|_2 \\ &\quad - D_8 \sum_{i=\tau_k}^{d_m} \mathbb{E}\|\theta_{k-i} - \theta_{k-d_m}\|_2 - D_9 \kappa \rho^{m-1},\end{aligned}$$

where $d_m \leq (K_0+1)m + K_0$, $D_6 := L_V C_\delta L_\psi$, $D_7 := C_p L_\lambda + (1+\gamma)L_V^2 C_\psi + 2L_V L_\lambda$, $D_8 := L_V C_p |\mathcal{A}| L_\pi$ and $D_9 := 2L_V C_p$.

Now we start to prove Theorem 4.

Proof: By following the derivation of (41), we have

$$\begin{aligned}&\mathbb{E}[J_\lambda(\theta_{k+1}) - J_\lambda(\theta_k)] \geq \\ &\alpha \underbrace{\mathbb{E}\left\langle \nabla J_\lambda(\theta_k), \left(\hat{\delta}(\hat{x}_{(k)}, \omega_{k-\tau_k}) - \hat{\delta}(\hat{x}_{(k)}, \omega^*_k)\right)\psi_{\theta_{k-\tau_k}}(\hat{s}_{(k)}, \hat{a}_{(k)})\right\rangle}_{I_1}\end{aligned}$$

$$+\alpha\underbrace{\mathbb{E}\left\langle\nabla J_\lambda(\theta_k),\hat{\delta}(\hat{x}_{(k)},\omega_k^*)\psi_{\theta_{k-\tau_k}}(\hat{s}_{(k)},\hat{a}_{(k)})+\lambda\nabla R(\theta_{k-\tau_k})\right\rangle}_{I_2}$$

$$-\frac{L_\lambda}{2}C_p^2\alpha^2. \tag{71}$$

The item I_1 can be bounded by following (46) as

$$I_1\geq-\frac{1}{2}\mathbb{E}\|\nabla J_\lambda(\theta_k)\|_2^2-4C_\psi^2C_\delta^2K_0^2\beta^2-4C_\psi^2\mathbb{E}\|\omega_k-\omega_k^*\|_2^2. \tag{72}$$

Next we consider I_2. We first decompose it as

$$\begin{aligned}
&I_2\\
&=\mathbb{E}\left\langle\nabla J_\lambda(\theta_k),\hat{\delta}(x_{(k)},\omega_k^*)\psi_{\theta_{k-\tau_k}}(s_{(k)},a_{(k)})+\lambda\nabla R(\theta_{k-\tau_k})\right\rangle\\
&=\underbrace{\mathbb{E}\left\langle\nabla J_\lambda(\theta_k),\left(\hat{\delta}(x_{(k)},\omega_k^*)-\delta(x_{(k)},\theta_k)\right)\psi_{\theta_{k-\tau_k}}(s_{(k)},a_{(k)})\right\rangle}_{I_2^{(1)}}\\
&+\underbrace{\mathbb{E}\left\langle\nabla J_\lambda(\theta_k),\delta(x_{(k)},\theta_k)\psi_{\theta_{k-\tau_k}}(s_{(k)},a_{(k)})-\nabla J(\theta_k)\right\rangle}_{I_2^{(2)}}\\
&+\underbrace{\mathbb{E}\left\langle\nabla J_\lambda(\theta_k),\lambda\nabla R(\theta_{k-\tau_k})-\lambda\nabla R(\theta_k)\right\rangle}_{I_2^{(3)}}\\
&+\mathbb{E}\|\nabla J_\lambda(\theta_k)\|_2^2.
\end{aligned} \tag{73}$$

For some $m\in\mathbb{N}^+$, define $M:=(K_0+1)m+K_0$. Following Lemma 4, for some $d_m\leq M$ and positive constants D_2,D_3,D_4,D_5, $I_2^{(1)}$ can be bounded as

$$\begin{aligned}
&I_2^{(1)}\\
&=\mathbb{E}\left\langle\nabla J_\lambda(\theta_k),\left(\hat{\delta}(x_{(k)},\omega_k^*)-\delta(x_{(k)},\theta_k)\right)\psi_{\theta_{k-\tau_k}}(s_{(k)},a_{(k)})\right\rangle\\
&\geq-D_2\mathbb{E}\|\theta_{k-\tau_k}-\theta_{k-d_m}\|_2-D_3\mathbb{E}\|\theta_k-\theta_{k-d_m}\|_2\\
&\quad-D_4\sum_{i=k-d_m}^{k-\tau_k}\mathbb{E}\|\theta_i-\theta_{k-d_m}\|_2-D_5\kappa\rho^{m-1}\\
&\quad-L_VC_\psi(1+\gamma)\epsilon_{\text{app}}\\
&\geq-D_2(d_m-\tau_k)C_p\alpha-D_3d_mC_p\alpha-D_4(d_m-\tau_k)^2C_p\alpha\\
&\quad-D_5\kappa\rho^{m-1}-(1+\gamma)L_VC_\psi\epsilon_{\text{app}},
\end{aligned} \tag{74}$$

where the derivation of the last inequality is similar to that of (67). By setting $m=m_K$ in (74), and following the fact that $d_{m_K}\leq M_K$ and $\tau_k\geq 0$, we have

$$\begin{aligned}
I_2^{(1)}\geq&-D_2M_KC_p\alpha-D_3M_KC_p\alpha-D_4M_K^2C_p\alpha\\
&-D_5\kappa\rho^{m_K-1}-(1+\gamma)L_VC_\psi\epsilon_{\text{app}}\\
=&-\left((D_2+D_3)C_pM_K+D_4C_pM_K^2\right)\alpha\\
&-D_5\kappa\rho^{m_K-1}-(1+\gamma)L_VC_\psi\epsilon_{\text{app}}\\
\geq&-\left((D_2+D_3)C_pM_K+D_4C_pM_K^2\right)\alpha\\
&-D_5\alpha-(1+\gamma)L_VC_\psi\epsilon_{\text{app}},
\end{aligned} \tag{75}$$

where the last inequality is due to the definition of m_K.

Following Lemma 5, for some positive constants D_6,D_7 and D_8, we bound $I_2^{(2)}$ as

$$\begin{aligned}
I_2^{(2)}=&\mathbb{E}\left\langle\nabla J_\lambda(\theta_k),\delta(x_{(k)},\theta_k)\psi_{\theta_{k-\tau_k}}(s_{(k)},a_{(k)})-\nabla J(\theta_k)\right\rangle\\
\geq&-D_6\mathbb{E}\|\theta_{k-\tau_k}-\theta_{k-d_m}\|_2-D_7\mathbb{E}\|\theta_k-\theta_{k-d_m}\|_2\\
&-D_8\sum_{i=\tau_k}^{d_m}\mathbb{E}\|\theta_{k-i}-\theta_{k-d_m}\|_2-D_9\kappa\rho^{m-1}.
\end{aligned}$$

Similar to the derivation of (75), we have

$$I_2^{(2)}\geq-\left(D_6C_pM_K+D_7C_pM_K+D_8C_pM_K^2\right)\alpha-D_9\alpha. \tag{76}$$

Term $I_2^{(3)}$ can be bounded as

$$\begin{aligned}
I_2^{(3)}&\geq-\lambda L_V\|\nabla R(\theta_k)-\nabla R(\theta_{k-\tau_k})\|_2\\
&\geq-\lambda L_VL_\psi\|\theta_k-\theta_{k-\tau_k}\|_2\\
&\geq-\lambda L_VL_\psi K_0C_p\alpha.
\end{aligned} \tag{77}$$

Collecting the lower bounds of $I_2^{(1)}$, $I_2^{(2)}$ and $I_2^{(3)}$ yields

$$I_2\geq-D_K\alpha-(1+\gamma)L_VC_\psi\epsilon_{\text{app}}+\mathbb{E}\|\nabla J_\lambda(\theta_k)\|_2^2, \tag{78}$$

where we define $D_K:=C_p(D_4+D_8)M_K^2+C_p(D_2+D_3+D_6+D_7)M_K+\lambda L_VL_\psi K_0C_p+D_5+D_9$ for brevity.

Substituting 72 and 78 into (71) yields

$$\begin{aligned}
&\mathbb{E}[J_\lambda(\theta_{k+1})-J_\lambda(\theta_k)]\\
&\geq\frac{\alpha}{2}\mathbb{E}\|\nabla J_\lambda(\theta_k)\|_2^2-4C_\psi^2\alpha\mathbb{E}\|\omega_k-\omega_k^*\|_2^2-4C_\psi^2C_\delta^2K_0^2\alpha\beta^2\\
&\quad-2L_VC_\psi\epsilon_{\text{app}}\alpha-D_K\alpha^2.
\end{aligned} \tag{79}$$

Choose step size $\alpha=K^{-\frac{3}{5}}$, $\beta=K^{-\frac{2}{5}}$. With $D_K=\mathcal{O}(M_K^2)=\mathcal{O}(K_0^2\log^2 K)$, the last inequality implies

$$\frac{1}{K}\sum_{k=1}^K\mathbb{E}\|\nabla J_\lambda(\theta_k)\|_2^2 \tag{80}$$

$$=\mathcal{O}\left(\frac{1}{K}\sum_{k=1}^K\mathbb{E}\|\omega_k-\omega_k^*\|_2^2\right)+\mathcal{O}\left(\frac{K_0^2\log^2 K}{K^{\frac{3}{5}}}\right)+\mathcal{O}\left(\epsilon_{\text{app}}\right). \tag{81}$$

This completes the proof. □

REFERENCES

[1] A. Agarwal and J. C. Duchi, "Distributed delayed stochastic optimization," in *Proc. 24th Int. Conf. Neural Inf. Process. Syst.*, 2011, pp. 873–881.

[2] A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan, "Optimality and approximation with policy gradient methods in Markov decision processes," in *Proc. 30th Conf. Learn. Theory*, 2020, pp. 64–66.

[3] M. Assran, J. Romoff, N. Ballas, J. Pineau, and M. Rabbat, "Gossip-based actor-learner architectures for deep reinforcement learning," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 13320–13330.

[4] M. Babaeizadeh, I. Frosio, S. Tyree, J. Clemons, and J. Kautz, "Reinforcement learning through asynchronous advantage actor-critic on a GPU," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–12.

[5] J. Baxter and P. L. Bartlett, "Infinite-horizon policy-gradient estimation," *J. Artif. Intell. Res.*, vol. 15, pp. 319–350, 2001.

[6] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Upper Saddle River, NJ, USA: Prentice-Hall, 1989.

[7] J. Bhandari and D. Russo, "Global optimality guarantees for policy gradient methods," 2022, *arXiv:1906.01786*.

[8] J. Bhandari and D. Russo, "On the linear convergence of policy gradient methods for finite MDPs," in *Proc. 24th Int. Conf. Artif. Intell. Statist.*, 2021, pp. 2386–2394.

[9] J. Bhandari, D. Russo, and R. Singal, "A finite time analysis of temporal difference learning with linear function approximation," in *Proc. Conf. Learn. Theory*, 2018, pp. 1691–1692.

[10] S. Bhatnagar, R. S. Sutton, M. Ghavamzadeh, and M. Lee, "Natural actor critic algorithms," *Automatica*, vol. 45, pp. 2471–2482, 2009.

[11] V. Borkar and V. Konda, "The actor-critic algorithm as multi-time-scale stochastic approximation," *Sadhana*, vol. 22, no. 4, pp. 525–543, 1997.

[12] J. Cervino, J. A. Bazerque, M. Calvo-Fullana, and A. Ribeiro, "Multi-task reinforcement learning in reproducing kernel hilbert spaces via cross-learning," *IEEE Trans. Signal Process.*, vol. 69, pp. 5947–5962, 2021.

[13] S. Chai and V. K. N. Lau, "Joint rate and power optimization for multimedia streaming in wireless fading channels via parametric policy gradient," *IEEE Trans. Signal Process.*, vol. 67, no. 17, pp. 4570–4581, Sep. 2019.

[14] S. Chai and V. K. N. Lau, "Online trajectory and radio resource optimization of cache-enabled UAV wireless networks with content and energy recharging," *IEEE Trans. Signal Process.*, vol. 68, pp. 1286–1299, 2020.

[15] T. Chen, K. Zhang, G. B. Giannakis, and T. Başar, "Communication-efficient distributed reinforcement learning," 2021, *arXiv:1812.03239*.

[16] F. Christianos, L. Schäfer, and S. Albrecht, "Shared experience actor-critic for multi-agent reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 10707–10717.

[17] Dgriff, "Pytorch implementation of a3c," 2018. [Online]. Available: https://github.com/dgriff777/rl_a3c_pytorch

[18] K. Doya, "Reinforcement learning in continuous time and space," *Neural Comput.*, vol. 12, no. 1, pp. 219–245, 2000.

[19] Y. El-Laham and M. F. Bugallo, "Policy gradient importance sampling for Bayesian inference," *IEEE Trans. Signal Process.*, vol. 69, pp. 4245–4256, 2021.

[20] L. Espeholtet al., "Impala: Scalable distributed deep-RL with importance weighted actor-learner architectures," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1407–1416.

[21] H. R. Feyzmahdavian, A. Aytekin, and M. Johansson, "An asynchronous mini-batch algorithm for regularized stochastic optimization," *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 740–3754, Dec. 2016.

[22] Z. Fu, Z. Yang, and Z. Wang, "Single-timescale actor-critic provably finds globally optimal policy," 2020, *arXiv:2008.00483*.

[23] M. Hong, H.-T. Wai, Z. Wang, and Z. Yang, "A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic," 2022, *arXiv:2007.05170*.

[24] S. Kar, J. Moura, and V. Poor, "QD-learning: A collaborative distributed strategy for multi-agent reinforcement learning through consensus innovations," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1848–1862, Apr. 2013.

[25] V. Konda, "Actor-critic algorithms," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2002.

[26] V. Konda and V. Borkar, "Actor-critic–type learning algorithms for Markov decision processes," *SIAM J. Control Optim.*, vol. 38, no. 1, 2019 pp. 94–123, 1999.

[27] H. Kumar, A. Koppel, and A. Ribeiro, "On the sample complexity of actor-critic method for reinforcement learning with function approximation," 2023, *arXiv:1910.08412*.

[28] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5336–5346.

[29] X. Lian, H. Zhang, C. Hsieh, Y. Yijun Huang, and J. Liu, "A comprehensive linear speedup analysis for asynchronous stochastic parallel optimization from zeroth-order to first-order," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3062–3070.

[30] T. P. Lillicrapet al., "Continuous control with deep reinforcement learning," in *Proc. Int. Conf. Learn. Representations*, 2016.

[31] J. Mei, C. Xiao, C. Szepesvari, and D. Schuurmans, "On the global convergence rates of softmax policy gradient methods," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 6820–6829.

[32] V. Mnihet al., "Asynchronous methods for deep reinforcement learning.," in *Proc. Int. Conf. Mach. Learn.*, 2016.

[33] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.

[34] A. Mokhtari, A. Koppel, M. Takáč, and A. Ribeiro, "A class of parallel doubly stochastic algorithms for large-scale learning," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 4718–4768, 2020.

[35] A. Nair et al., "Massively parallel methods for deep reinforcement learning," 2015, *arXiv:1507.04296*.

[36] S. Qiu, Z. Yang, J. Ye, and Z. Wang, "On the finite-time convergence of actor-critic algorithm," in *Proc. IEEE Optim. Found. Reinforcement Learn. Workshop Adv. Neural Inf. Process. Syst.*, 2019.

[37] G. Qu, Y. Lin, A. Wierman, and N. Li, "Scalable multi-agent reinforcement learning for networked systems with average reward," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 2074–2086.

[38] B. Recht, C. Re, S. Wright, and F. Niu, "Hogwild!: A lock-free approach to parallelizing stochastic gradient descent," in *Proc. 24th Int. Conf. Neural Inf. Process. Syst.*, 2011, pp. 693–701.

[39] A. Sadeghi, F. Sheikholeslami, and G. B. Giannakis, "Optimal and scalable caching for 5G using reinforcement learning of space-time popularities," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 180–190, Feb. 2018.

[40] H. Shen, K. Zhang, M. Hong, and T. Chen, "Asynchronous advantage actor critic: Non-asymptotic analysis and linear speedup," 2022, *arXiv:2012.15511*.

[41] A. Stooke and P. Abbeel, "Accelerated methods for deep reinforcement learning," 2019, *arXiv:1803.02811*.

[42] T. Sun, R. Hannah, and W. Yin, "Asynchronous coordinate descent under more realistic assumptions," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6183–6191.

[43] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Mach. Learn.*, vol. 3, pp. 9–44, 1988.

[44] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[45] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation.," in *Proc. 12th Int. Conf. Neural Inf. Process. Syst.*, 1999, pp. 1057–1063.

[46] L. Wang, Q. Cai, Z. Yang, and Z. Wang, "Neural policy gradient methods: Global optimality and rates of convergence," 2019, *arXiv:1909.01150*.

[47] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, no. 3/4, pp. 229–256, 1992.

[48] T. Wu, K. Yuan, Q. Ling, W. Yin, and A. Sayed, "Decentralized consensus optimization with asynchrony and delays," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 2, pp. 293–307, Jun. 2018.

[49] Y. Wu, W. Zhang, P. Xu, and Q. Gu, "A finite time analysis of two time-scale actor critic methods," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 17617–17628.

[50] Z. Wu, H. Shen, T. Chen, and Q. Ling, "Byzantine-resilient decentralized policy evaluation with linear function approximation," *IEEE Trans. Signal Process.*, vol. 69, pp. 3839–3853, 2021.

[51] T. Xu, Z. Wang, and Y. Liang, "Improving sample complexity bounds for (natural) actor-critic algorithms," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 4358–4369.

[52] T. Xu, Z. Wang, and Y. Liang, "Non-asymptotic convergence analysis of two time-scale (natural) actor-critic algorithms," 2020, *arXiv:2005.03557*.

[53] Z. Yang, K. Zhang, M. Hong, and T. Başar, "A finite sample analysis of the actor-critic algorithm," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 2759–2764.

[54] J. Zhang, J. Kim, B. Donoghue, and S. Boyd, "Sample efficient reinforcement learning with reinforce," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 10887–10895.

[55] K. Zhang, A. Koppel, H. Zhu, and T. Başar, "Global convergence of policy gradient methods to (almost) locally optimal policies," *SIAM J. Control Optim.*, 2019, pp. 3586–3612.

[56] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Başar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 9340–9371.

[57] S. Zou, T. Xu, and Y. Liang, "Finite-sample analysis for SARSA with linear function approximation," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 8668–8678.