Finite-Time Analysis of Decentralized Single-Timescale Actor-Critic

Anonymous authors
Paper under double-blind review

Abstract

Decentralized Actor-Critic (AC) algorithms have been widely utilized for multi-agent reinforcement learning (MARL) and have achieved remarkable success. Apart from its empirical success, the theoretical convergence property of decentralized AC algorithms is largely unexplored. Most of the existing finite-time convergence results are derived based on either double-loop update or two-timescale step sizes rule. In practice, the *single-timescale* update is widely utilized, where actor and critic are updated in an alternating manner with step sizes being of the same order. In this work, we study a decentralized *single-timescale* AC algorithm. Theoretically, using linear approximation for value and reward estimation, we show that the algorithm has sample complexity of $\tilde{\mathcal{O}}(\varepsilon^{-2})$ under Markovian sampling, which matches the optimal complexity with double-loop implementation (here, $\tilde{\mathcal{O}}$ hides a logarithmic term). The central to establishing our complexity results is *the hidden smoothness of the optimal critic variable* we revealed. We also provide a local action privacy-preserving version of our algorithm and its analysis. Finally, we conduct experiments to show the superiority of our algorithm over the existing decentralized AC algorithms.

1 Introduction

Multi-agent reinforcement learning (MARL) (Littman, 1994; Vinyals et al., 2019) has been successful in various models of multi-agent systems, such as robotics (Lillicrap et al., 2015), autonomous driving (Yu et al., 2019), Go (Silver et al., 2017), etc. MARL has been extensively explored in the past decades; see, e.g., (Lowe et al., 2017; Omidshafiei et al., 2017; Zhang et al., 2021; Son et al., 2019; Espeholt et al., 2018; Rashid et al., 2018). These works either focus on the setting where a central controller is available, or assuming a common reward function for all agents. Among the many cooperative MARL settings, the work (Zhang et al., 2018) proposes the fully decentralized MARL with networked agents. In this setting, each agent maintains a private heterogeneous reward function, and agents can only access local/neighboring information through communicating with its neighboring agents on the network. Then, the objective of all agents is to jointly maximize the average long-term reward through interacting with environment modeled by multi-agent Markov decision process (MDP). They proposed the decentralized Actor-Critic (AC) algorithm to solve this MARL problem, and showed its impressive performance. However, the theoretical convergence properties of such class of decentralized AC algorithms are largely unexplored; see (Zhang et al., 2021) for a comprehensive survey. In this work, our goal is to establish the finite-time convergence results under this fully decentralized MARL setting. We first review some recent progresses on this line of research below.

Related works and motivations. The first fully decentralized AC algorithm with provable convergence guarantee was proposed by (Zhang et al., 2018), and they achieved asymptotic convergence results under two-time scale step sizes, which requires actor's step sizes to diminish in a faster scale than the critic's step sizes. The sample complexities of decentralized AC were established recently. In particular, (Chen et al., 2022) and (Hairi et al., 2022) independently proposed two communication efficient decentralized AC algorithms with optimal sample complexity of $\mathcal{O}(\varepsilon^{-2}\log(\varepsilon^{-1}))$ under Markovian sampling scheme. Nevertheless, their analysis are based on *double-loop* implementation, where each policy optimization step follows a nearly accurate critic optimization step (a.k.a. policy evaluation), i.e., solving the critic optimization subproblem to ε-accuracy. Such a double-loop scheme requires careful tuning of two additional hyper-parameters, which are the batch size and inner loop size. In particular, the batch size and inner loop size need to be of order $\mathcal{O}(\varepsilon^{-1})$ and $\mathcal{O}(\log(\varepsilon^{-1}))$ in order to achieve their sample complexity results, respectively. In practice, single-loop algorithmic framework is often utilized, where one updates the actor and critic in an alternating manner by performing only one algorithmic iteration for both of the two subproblems; see, e.g., (Schulman et al., 2017; Lowe et al., 2017; Lin et al., 2019; Zhang et al., 2020). The work (Zeng et al., 2021) proposed a new decentralized AC algorithm based on such a single-loop alternative update. Nevertheless, they have to adopt *two-timescale* step sizes rule to ensure convergence, which requires actor's step sizes to diminish in a faster scale than the critic's step sizes. Due to the separation of the step sizes, the critic optimization sub-problem is solved exactly when the number of iterations tends to ∞. Such a restriction on the step size will slow down the convergence speed of the algorithm. As a consequence, they only obtain sub-optimal sample complexity of $\mathcal{O}(\varepsilon^{-\frac{5}{2}})$. In practice, most algorithms are implemented with *single-timescale* step size rule, where the step sizes for actor and critic updates are of the same order. Though there are some theoretical achievements for single-timescale update in other areas such as TDC (Wang et al., 2021) and bi-level optimization (Chen et al., 2021a), similar theoretical understanding under AC setting is largely unexplored.

Indeed, even when reducing to single-agent setting, the convergence property of single-timescale AC algorithm is not well established. The works (Fu et al., 2021; Guo et al., 2021) established the finite-time convergence result under a special single-timescale implementation, where they attained the sample complexity of $\mathcal{O}(\varepsilon^{-2})$. However, their analysis is based on an algorithm where the critic optimization step is formulated as a least-square temporal difference (LSTD) at each iteration, where they need to sample the transition tuples for $\tilde{\mathcal{O}}(\varepsilon^{-1})$ times to form the data matrix in the LSTD problem. Then, they solve the LSTD problem in a closed-form fashion, which requires to invert a matrix of large size. Later, (Chen et al., 2021a) obtained the same sample complexity using TD(0) update for critic variables under i.i.d. sampling. Nonetheless, their analysis highly relies on the assumption that the Jacobian of the stationary distribution is Lipschitz continuous, which is not justified in their work.

The above observations motivate us to ask the following question:

Can we establish finite-time convergence result for decentralized AC algorithm with single-timescale step sizes rule?[1]

Main contributions. By answering this question positively, we have the following contributions:

- We design a fully decentralized AC algorithm, which employs a *single-timescale* step sizes rule and adopts Markovian sampling scheme. The proposed algorithm allows communication between agents for every K_c iterations with K_c being any integer lies in $[1, \mathcal{O}(\varepsilon^{-\frac{1}{2}})]$, rather than communicating at each iteration as adopted by previous single-loop decentralized AC algorithms (Zeng et al., 2021; Zhang et al., 2018).
- Using linear approximation for value and reward estimation, we establish the *finite-time* convergence result for such an algorithm under the standard assumptions. In particular, we show that the algorithm has the sample complexity of $\tilde{\mathcal{O}}(\varepsilon^{-2})$, which matches the optimal complexity up to a logarithmic term. In addition, we show that the logarithmic term can be removed under the i.i.d. sampling scheme. These convergence results are valid for all the above mentioned choices for K_c.
- To preserve the privacy of local actions, we propose a variant of our algorithm which utilizes noisy local rewards for estimating global rewards. We show that such an algorithm will maintain the optimal sample complexity at the expense of communicating at each iteration.

The underlying principle for obtaining the above convergence results is that we reveal *the hidden smoothness of the optimal critic variable*, so that we can derive an approximate descent on the averaged critic's optimal gap at each iteration. Consequently, we can resort to the classic convergence analysis for alternating optimization algorithms to establish the approximate ascent property of the overall optimization process, which leads to the final sample complexity results.

We remark that our convergence results are even new for single-agent AC algorithms under the setting of single-timescale step sizes rule.

---

[1] As convention in (Fu et al., 2021), when we use "single-timescale", it means we utilize a single-loop algorithmic framework with single-timescale step sizes rule.

Discussion on a concurrent work. We note that there is a concurrent work (Olshevsky & Gharesifard, 2022) which also analyzes the single-timescale AC algorithm and achieves similar complexity results. Their analysis is based on the small gain theorem, which is different from our analysis. These two analysis frameworks provide useful insights for the AC algorithm from different perspectives. (Olshevsky & Gharesifard, 2022) shows that the coupled expression on the errors of actor and critic can be fit into a non-linear small gain theorem framework, which bounds the actor's error by desired order. Our analysis reveal the hidden smoothness of the optimal critic variable so that approximate descent on the critic's objective can be achieved. Moreover, the two works consider different problem settings. (Olshevsky & Gharesifard, 2022) considers the single-agent setting while our analysis deals with more general fully decentralized setting. In addition, (Olshevsky & Gharesifard, 2022) analyzes the i.i.d. sampling scheme where agents are assumed to have access to transition tuples from the stationary distribution and the discounted state-visitation distribution. By contrast, our setting consider the practical Markovian sampling scheme, where the transition tuples are from the trajectory generated during the update of agents.

2 Preliminary

In this section, we introduce the problem formulation and the policy gradient theorem, which serves as the preliminary for the analyzed decentralzed AC algorithm.

Suppose there are multiple agents aiming to independently optimize a common global objective, and each agent can communicate with its neighbors through a network. To model the topology, we define the graph as $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes with $|\mathcal{N}| = N$ and $\mathcal{E}$ is the set of edges with $|\mathcal{E}| = E$. In the graph, each node represents an agent, and each edge represents a communication link. The interaction between agents follows the networked multi-agent MDP.

2.1 Markov decision process

A networked multi-agent MDP is defined by a tuple $(\mathcal{G}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, \mathcal{P}, \{r^i\}_{i\in[N]}, \gamma)$. $\mathcal{G}$ denotes the communication topology (the graph), $\mathcal{S}$ is the finite state space observed by all agents, $\mathcal{A}^i$ represents the finite action space of agent i. Let $\mathcal{A} := \mathcal{A}^1 \times \cdots \times \mathcal{A}^N$ denote the joint action space and $\mathcal{P}(s'|s,a) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$ denote the transition probability from any state $s \in \mathcal{S}$ to any state $s' \in \mathcal{S}$ for any joint action $a \in \mathcal{A}$. $r^i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the local reward function that determines the reward received by agent i given transition (s,a); $\gamma \in [0,1]$ is the discount factor.

For simplicity, we will use $a := [a^1, \cdots, a^N]$ to denote the joint action, and $\theta := [\theta^1, \cdots, \theta^N] \in \mathbb{R}^{d_\theta \times N}$ to denote joint parameters of all actors, with $\theta^i \in \mathbb{R}^{d_\theta}$. Note that different actors may have different number of parameters, which is assumed to be the same for our paper without loss of generality. The MDP goes as follows: For a given state s, each agent make its decision a^i based on its policy $a^i \sim \pi_{\theta^i}(\cdot|s)$. The state transits to the next state s' based on the joint action of all the agents: $s' \sim \mathcal{P}(\cdot|s,a)$. Then, each agent will receive its own reward $r^i(s,a)$. For the notation brevity, we assume that the reward function mapping is deterministic and does not depend on the next state without loss of generality. The stationary distribution induced by the policy π_θ and the transition kernel is denoted by $\mu_{\pi_\theta}(s)$.

Our objective is to find a set of policies that maximize the accumulated discounted mean reward received by agents

$$\theta^* = \arg\max_\theta J(\theta) := \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \bar{r}(s_k, a_k)\right]. \tag{1}$$

Here, k represents the time step. $\bar{r}(s_k, a_k) := \frac{1}{N}\sum_{i=1}^N r^i(s_k, a_k)$ is the mean reward among agents at time step k. The randomness of the expectation comes from the initial state distribution $\mu_0(s)$, the transition kernel $\mathcal{P}$, and the stochastic policy $\pi_{\theta^i}(\cdot|s)$.

2.2 Policy gradient Theorem

Under the discounted reward setting, the global state-value function, action-value function, and advantage function for policy set θ, state s, and action a, are defined as

$$\begin{aligned} V_{\pi_\theta}(s) &:= \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \bar{r}(s_k, a_k) | s_0 = s\right] \\ Q_{\pi_\theta}(s,a) &:= \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \bar{r}(s_k, a_k) | s_0 = s, a_0 = a\right] \\ A_{\pi_\theta}(s,a) &:= Q_{\pi_\theta}(s,a) - V_{\pi_\theta}(s). \end{aligned} \tag{2}$$

To maximize the objective function defined in (1), the policy gradient (Sutton et al., 2000) can be computed as follow

$$\nabla_\theta J(\theta) = \mathbb{E}_{s\sim d_{\pi_\theta}, a\sim\pi_\theta}\left[\frac{1}{1-\gamma} A_{\pi_\theta}(s,a)\psi_{\pi_\theta}(s,a)\right],$$

where $d_{\pi_\theta}(s) := (1-\gamma)\sum_{k=0}^{\infty}\gamma^k \mathbb{P}(s_k = s)$ is the discounted state visitation distribution under policy π_θ, and $\psi_{\pi_\theta}(s,a) := \nabla \log \pi_\theta(s,a)$ is the score function.

Following the derivation of (Zhang et al., 2018), the policy gradient for each agent under discounted reward setting can be expressed as

$$\nabla_{\theta^i} J(\theta) = \mathbb{E}_{s\sim d_{\pi_\theta}, a\sim\pi_\theta}\left[\frac{1}{1-\gamma} A_{\pi_\theta}(s,a)\psi_{\pi_{\theta^i}}(s,a^i)\right]. \tag{3}$$

3 Algorithms

3.1 Decentralized single-timescale actor-critic

Algorithm 1: Decentralized single-timescale AC (reward estimator version)

```
Initialize: Actor parameter $\theta_0$, critic parameter $\omega_0$, reward estimator parameter $\lambda_0$, initial state $s_0$.
for $k = 0, \cdots, K-1$ do
    Option 1: i.i.d. sampling:
    $s_k \sim \mu_{\theta_k}(s), a_k \sim \pi_{\theta_k}(\cdot|s_k), s_{k+1} \sim \mathcal{P}(\cdot|s_k, a_k)$.
    Option 2: Markovian sampling:
    $a_k \sim \pi_{\theta_k}(\cdot|s_k), s_{k+1} \sim \mathcal{P}(\cdot|s_k, a_k)$.

    Periodical consensus: Compute $\tilde{\omega}_k^i$ and $\tilde{\lambda}_k^i$ by (4) and (7).

    for $i = 0, \cdots, N$ in parallel do
        Reward estimator update: update $\lambda_{k+1}^i$ by (8).
        Critic update: Update $\omega_{k+1}^i$ by (5).
        Actor update: Update $\theta_{k+1}^i$ by (6).
    end for
end for
```

We introduce the decentralized single-timescale AC algorithm; see Algorithm 1. In the remaining parts of this section, we will explain the updates in the algorithm in details.

In fully-decentralized MARL, each agent can only observe its local reward and action, while trying to maximize the global reward (mean reward) defined in (1). The decentralized AC algorithm solves the problem by performing online updates in an alternative fashion. Specifically, we have N pairs of actor and critic. In order to maximize $J(\theta)$, each critic tries to estimate the *global* state-value function $V_{\pi_\theta}(s)$ defined in (2), and each actor then updates its policy parameter based on approximated policy gradient. We now provide more details about the algorithm.

Critics' update. We will use $\omega^i \in \mathbb{R}^{d_\omega}$ to denote the i_{th} critic's parameter and $\bar{\omega} := \frac{1}{N}\sum_{i=1}^N \omega^i$ to represent the averaged parameter of critic. The i_{th} critic approximates the global value function as $V_{\pi_\theta}(s) \approx \hat{V}_{\omega^i}(s)$.

As we will see, the critic's approximation error can be categorized into two parts, namely, the consensus error $\frac{1}{N}\sum_{i=1}^N \|\omega^i - \bar{\omega}\|$, which measures how close the critics' parameters are; and the approximation error $\|\bar{\omega} - \omega^*(\theta)\|$, which measures the approximation quality of averaged critic.

In order for critics to reach consensus, each critic exchanges its parameters with neighbors and perform the following update

$$\tilde{\omega}_k^i = \begin{cases} \sum_{j=1}^N W^{ij}\omega_k^j & \text{if } k \mod K_c = 0 \\ \omega_k^i & \text{otherwise.} \end{cases} \tag{4}$$

Here, K_c denotes the consensus frequency. The communication matrix $W \in \mathbb{R}^{n\times n}$ is usually determined artificially in practice and can be sparse, which means that the number of neighbors for each agent is much fewer than the total number of agents. Thus, the cost for each consensus step is usually much lower than a full synchronization.

To reduce the approximation error, we will perform the local TD(0) update (Tsitsiklis & Van Roy, 1997) as

$$\omega_{k+1}^i = \Pi_{R_\omega}(\tilde{\omega}_k^i + \beta_k g_c^i(\xi_k, \omega_k^i)), \tag{5}$$

where $\xi := (s, a, s')$ represents a transition tuple, $g_c^i(\xi,\omega) := \delta^i(\xi,\omega)\nabla\hat{V}_\omega(s)$ is the update direction, $\delta^i(\xi,\omega) := r^i(s,a) + \gamma\hat{V}_\omega(s') - \hat{V}_\omega(s)$ is the local temporal difference error (TD-error). β_k is the step size for critic at iteration k. Π_{R_ω} projects the parameter into a ball of radius of R_ω containing the optimal solution, which will be explained when discussing Assumptions 1 and 2.

Actors' update. We will use stochastic gradient ascent to update the policy's parameter, and the stochastic gradient is calculated based on policy gradient theorem in (3). The advantage function $A_{\pi_\theta}(s,a)$ can be estimated by

$$\delta(\xi,\theta) := \bar{r}(s,a) + \gamma V(s') - V(s),$$

with a sampled from $\pi_\theta(\cdot|s)$. However, to preserve the privacy of each agents, the local reward cannot be shared to other agents under the fully decentralized setting. Thus, the averaged reward $\bar{r}(s_k, a_k)$ is not directly attainable. Consequently, we need a strategy to approximate the averaged reward. In this paper, we will adopt the strategy proposed in (Zhang et al., 2018). In particular, each agent i will have a local reward estimator with parameter $\lambda^i \in \mathbb{R}^{d_\lambda}$, which estimates the global averaged reward as $\bar{r}(s_k, a_k) \approx \hat{r}_{\lambda^i}(s_k, a_k)$.

Thus, the update of the i_{th} actor is given by

$$\theta_{k+1}^i = \theta_k^i + \alpha_k\hat{\delta}(\xi_k, \omega_{k+1}^i, \lambda_{k+1}^i)\psi_{\pi_{\theta_k^i}}(s_k, a_k^i), \tag{6}$$

where $\hat{\delta}(\xi,\omega,\lambda) := \hat{r}_\lambda(s,a) + \gamma\hat{V}_\omega(s') - \hat{V}_\omega(s)$ is the approximated advantage function. α_k is the step size for actor's update at iteration k.

Reward estimators' update. Similar to critic, each reward estimator's approximation error can be decomposed into consensus error and the approximation error.

For each local reward estimator, we perform the consensus step to minimize the consensus error as

$$\tilde{\lambda}_k^i = \begin{cases} \sum_{j=1}^N W^{ij}\lambda_k^j & \text{if } k \mod K_c = 0 \\ \lambda_k^i & \text{otherwise.} \end{cases} \tag{7}$$

To reduce the approximation error, we perform a local update of stochastic gradient descent.

$$\lambda_{k+1}^i = \Pi_{R_\lambda}(\tilde{\lambda}_k^i + \eta_k g_r^i(\xi_k, \lambda_k^i)), \tag{8}$$

Algorithm 2: Decentralized single-timescale AC (noisy reward version)

```
Initialize: Actor parameter $\theta_0$, critic parameter $\omega_0$, initial state $s_0$.
for $k = 0, \cdots, K-1$ do
    Option 1: i.i.d. sampling:
    $s_k \sim \mu_{\theta_k}(s), a_k \sim \pi_{\theta_k}(\cdot|s_k), s_{k+1} \sim \mathcal{P}(\cdot|s_k, a_k)$.
    Option 2: Markovian sampling:
    $a_k \sim \pi_{\theta_k}(\cdot|s_k), s_{k+1} \sim \mathcal{P}(\cdot|s_k, a_k)$.

    Periodical consensus: Compute $\tilde{\omega}_k^i$ by (4).

    for $i = 0, \cdots, N$ in parallel do
        Global reward estimation: estimate $\bar{r}_k(s_k, a_k)$ by (9).
        Critic update: Update $\omega_{k+1}^i$ by (5).
        Actor update: Update $\theta_{k+1}^i$ by (10).
    end for
end for
```

where $g_r^i(\xi, \lambda) := (r^i(s,a) - \hat{r}_\lambda(s,a))\nabla\hat{r}_\lambda(s,a)$ is the update direction. η_k is the step size for reward estimator at iteration k. Note the calculation of $g_r^i(\xi, \lambda)$ does not require the knowledge of s'; we use ξ in (8) just for notation brevity. Similar to critic's update, Π_{R_λ} projects the parameter into a ball of radius of R_λ containing the optimal solution.

In our Algorithm 1, we will use the same order for α_k, β_k, and η_k and hence, our algorithm is in *single-timescale*.

Linear approximation for analysis. In our analysis, we will use linear approximation for both critic and reward estimator variables, i.e. $\hat{V}_\omega(s) := \phi(s)^T\omega; \hat{r}_\lambda(s,a) := \varphi(s,a)^T\lambda$, where $\phi(s) : \mathcal{S} \to \mathbb{R}^{d_\omega}$ and $\varphi(s,a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_\lambda}$ are two feature mappings, whose property will be specified in the discussion of Assumption 1.

Remarks on sampling scheme. The unbiased update for critic and actor variables requires sampling from μ_{π_θ} and d_{π_θ}, respectively. However, in practical implementations, states are usually collected from an online trajectory (Markovian sampling), whose distribution is generally different for μ_{π_θ} and d_{π_θ}. Such a distribution mismatch will inevitably cause biases during the update of critic and actor variables. One has to bound the corresponding error terms when analyzing the algorithm.

3.2 Variant for preserving local action

Note that in Algorithm 1, the reward estimators need the knowledge of joint actions in order to estimate the global rewards. Inspired by (Chen et al., 2022), we further propose a variant of Algorithm 1 to preserve the privacy of local actions. It estimates the global rewards by communicating noisy local rewards. As a trade-off, the approach requires $\mathcal{O}(\log(\varepsilon^{-1}))$ communication rounds for each iteration; see Algorithm 2.

Let r_k^i represents $r_k^i(s_k, a_k)$ for brevity. The reward estimation process goes as follow: for each agent i, we first produce a noisy local reward $\tilde{r}_k^i = r_k^i(1+z)$, with $z \sim \mathcal{N}(0, \sigma^2)$. Thus, the noise level is controlled by the variance σ^2, which is chosen artificially. To estimate the global reward, each agent i first initialize the estimation as $\tilde{r}_{t,0}^i = \tilde{r}_t^i$. Then, each agent i perform the following consensus step for K_r times, i.e.

$$\tilde{r}_{t,l+1}^i = \sum_{j=1}^{N} W^{ij}\tilde{r}_{t,l}^i, \quad l = 0, 1, \cdots, K_r - 1. \tag{9}$$

The reward $\tilde{r}_{k,K_r}^i$ will be used for estimating the global reward for agent i at k_{th} iteration. As we will see, the error $|\tilde{r}_{t,l+1}^i - \frac{1}{N}\sum_{i=1}^N \tilde{r}_k^i|$ will converge to 0 linearly. Hence, to reduce the error to ε, we need $K_r = \mathcal{O}(\log(\varepsilon^{-1}))$ rounds of communications for each iteration. As we can see, when the noise level σ^2 increases, the local reward's privacy will be strengthen. In the meantime, the variance of the estimated global reward will be increased. Based on the estimated global reward, the i_{th} actor's update is given by

$$\theta^i_{k+1} = \theta^i_k + \alpha_k(\tilde{r}^i_{k,K_r} + \gamma \hat{V}_\omega(s') - \hat{V}_\omega(s))\psi_{\pi_{\theta^i_k}}(s_k, a^i_k). \tag{10}$$

4 Main results

In this section, we first introduce the technical assumptions used for our analysis, which are standard in the literature. Then, we present the convergence results for both actor and critic variables under i.i.d. sampling and Markovian sampling.

4.1 Assumptions

Assumption 1 (bounded rewards and feature vectors)**.** *All the local rewards are uniformly bounded, i.e., there exists a positive constant* $r_{\max}$ *such that* $|\,r^i(s,a)\,| \le r_{\max}$*, for all feasible* (s,a) *and* $i \in [N]$*. The norm of feature vectors are bounded such that for all* $s \in \mathcal{S},\ a \in \mathcal{A}$, $\|\phi(s)\| \le 1, \|\varphi(s,a)\| \le 1.$[2]

Assumption 1 is standard and commonly adopted; see, e.g., (Bhandari et al., 2018; Xu et al., 2020; Zeng et al., 2021; Shen et al., 2020; Qiu et al., 2019). This assumption can be achieved via normalizing the feature vectors.

Assumption 2 (negative definiteness of $A_{\theta,\phi}$ and $A_{\theta,\varphi}$)**.** *There exists two positive constants* $\lambda_\phi, \lambda_\varphi$ *such that for all policy* θ*, the following two matrices are negative definite*

$$A_{\theta,\phi} := \mathbb{E}_{s\sim\mu_\theta(s)}[\phi(s)(\gamma\phi(s')^T - \phi(s)^T)]$$
$$A_{\theta,\varphi} := \mathbb{E}_{s\sim\mu_\theta(s),a\sim\pi_\theta(\cdot|s)}[-\varphi(s,a)\varphi(s,a)^T],$$

with $\lambda_{\max}(A_{\theta,\phi}) \le \lambda_\phi, \lambda_{\max}(A_{\theta,\varphi}) \le \lambda_\varphi$*, where* $\lambda_{\max}(\cdot)$ *represents the largest eigenvalue.*

Assumption 2 will be satisfied when $\inf_{\theta,s,a} \pi_\theta(a|s) \ge c,\ \forall \theta, s, a$ for some positive constant c (see Proposition 3.1 of (Olshevsky & Gharesifard, 2022) for the proof). Thus, it can be understood as an exploration assumption on policy θ. This assumption is widely seen in analysis of AC algorithms; see e.g. (Shen et al., 2020; Xu & Liang, 2021; Zeng et al., 2021). Together with Assumption 1, we can show that the norm of $\omega^*(\theta)$ and $\lambda^*(\theta)$ are bounded by some positive constant, which justifies the projection steps; see Appendix C for the choice of the projection radius.

Assumption 3 (Lipschitz properties of policy)**.** *There exists constants* C_ψ, L_ψ, L_π *such that for all* $\theta, \theta', s \in \mathcal{S}$ *and* $a \in \mathcal{A}$*, we have (1).* $|\pi_\theta(a|s) - \pi_{\theta'}(a|s)| \le L_\pi\|\theta - \theta'\|;$ *(2).* $\|\psi_\theta(s,a) - \psi_{\theta'}(s,a)\| \le L_\psi\|\theta - \theta'\|;$ *(3).* $\|\psi_\theta(s,a)\| \le C_\psi$*.*

Assumption 3 is common for analyzing policy-based algorithms; see, e.g., (Xu et al., 2019; Wu et al., 2020; Hairi et al., 2022). The assumption ensures the smoothness of objective function $J(\theta)$. It holds for policy classes such as tabular softmax policy (Agarwal et al., 2020), Gaussian policy (Doya, 2000), and Boltzman policy (Konda & Borkar, 1999).

Assumption 4 (irreducible and aperiodic Markov chain)**.** *The Markov chain under* π_θ *and transition kernel* $\mathcal{P}(\cdot|s,a)$ *is irreducible and aperiodic for any* θ*.*

Assumption 4 is a standard assumption, which holds for any uniformly ergodic Markov chains and any time-homogeneous Markov chains with finite-state space. It ensures the geometric convergence to the stationary distribution, formally, there exists constants $\kappa > 0$ and $\rho \in (0,1)$ such that

$$\sup_{s\in\mathcal{S}} d_{TV}(\mathbb{P}(s_k \in \cdot|s_0 = s, \pi_\theta), \mu_\theta) \le \kappa\rho^k,\ \forall k.$$

Assumption 5 (doubly stochastic weight matrix)**.** *The communication matrix* W *is doubly stochastic, i.e. each column/row sum up to 1. Moreover, the second largest singular value* ν *is smaller than 1.*

Assumption 5 is a common assumption in decentralized optimization and multi-agent reinforcement learning; see, e.g., (Sun et al., 2020; Chen et al., 2021b; 2022). It ensures the convergence of consensus error for critic and reward estimator variables.

---

[2] Through out the paper, we will use $\|\cdot\|$ to represent the Euclidean norm for vectors and Frobenius norm for matrices.

4.2 Sample complexity for Algorithm 1

Theorem 1 (sample complexity under Markovian sampling)**.** *Suppose Assumptions 1-5 hold. Consider the update of Algorithm 1 under Markovian sampling. Let* $\alpha_k = \frac{\bar{\alpha}}{\sqrt{K}}$ *for some positive constant* $\bar{\alpha}$*,* $\beta_k = \frac{C_9}{2\lambda_\phi}\alpha_k$*,* $\eta_k = \frac{C_{10}}{2\lambda_\varphi}\alpha_k$*, and* $K_c \le \mathcal{O}(K^{1/4})$*, where* K *is the total number of iterations. Then, we have*

$$\frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{N}\mathbb{E}\left[\left\|\omega_k^i - \omega^*(\theta_k)\right\|^2\right] \le \mathcal{O}\left(\frac{\log^2 K}{\sqrt{K}}\right)$$

$$\frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{N}\mathbb{E}\left[\left\|\nabla_{\theta^i} F(\theta_k)\right\|^2\right] \le \mathcal{O}\left(\frac{\log^2 K}{\sqrt{K}}\right) + \mathcal{O}\left(\varepsilon_{app} + \varepsilon_{sp}\right), \tag{11}$$

where C_9, C_{10} *are positive constants defined in proof.*

The proof of Theorem 1 can be found in Appendix D.1. It establishes the iteration complexity of $\mathcal{O}(\log^2 K/\sqrt{K})$, or equivalently, sample complexity of $\tilde{\mathcal{O}}(\varepsilon^{-2})$ for Algorithm 1. Note that actors, critics, and reward estimators use the step size of the same order. The rate matches the state-of-the-art sample complexity of decentralized AC algorithms up to a logarithmic term, which are implemented in double-loop fashion (Hairi et al., 2022; Chen et al., 2022). The approximation error is defined as

$$\varepsilon_{app} := \max_{\theta, a}\mathbb{E}_{s\sim\mu_\theta}\left[\left|V_{\pi_\theta}(s) - \hat{V}_{\omega^*(\theta)}(s)\right|^2 + \left|\bar{r}(s,a) - \hat{r}_{\lambda^*(\theta)}(s,a)\right|^2\right]. \tag{12}$$

The error ε_{app} captures the approximation power of critic and reward estimator. When using function approximation, such an error is inevitable. Similar terms also appear in the literature (see e.g., (Xu et al., 2020; Agarwal et al., 2020; Qiu et al., 2019)). ε_{app} becomes zero in tabular case. The error ε_{sp} represents the mismatch between the discounted state visitation distribution d_{π_θ} and stationary distribution μ_{π_θ}. It is defined as

$$\varepsilon_{sp} := 4C_\theta^2\left(\log_\rho \kappa^{-1} + \frac{1}{\rho}\right)^2(1-\gamma)^2.$$

By policy gradient theorem (3), the states should be sampled from discounted state visitation distribution in order to attain unbiased estimation of policy gradient. Nevertheless, the state distribution converges to stationary distribution μ_{π_θ} due to Markov chain's mixing, which inevitably introduces the sampling error ε_{sp}; see (Zeng et al., 2021; Shen et al., 2020) for similar term. When γ is close to 1, the error becomes small. This is because d_{π_θ} approaches to μ_{π_θ} when γ goes to 1. In the literature, some works assume that sampling from d_{π_θ} is permitted, thus eliminate this error; see e.g., (Chen et al., 2021a).

Remark on the convergence under i.i.d. sampling. Under the i.i.d. sampling scheme, state can be directly sampled from μ_{π_θ} and d_{π_θ}. In this case, the logarithmic term caused by the Markovian mixing time, and the error ε_{sp} caused by the distribution mismatch, can be avoided. Hence, the iteration complexity of $\mathcal{O}(1/\sqrt{K})$, or equivalently, sample complexity of $\mathcal{O}(\varepsilon^{-2})$ will be obtained.

4.3 Sample complexity for Algorithm 2

Theorem 2. *Suppose Assumptions 1-5 hold. Consider the update of Algorithm 2 under Markovian sampling. Let* $\alpha_k = \frac{\bar{\alpha}}{\sqrt{K}}$ *for some positive constant* $\bar{\alpha}$*,* $\beta_k = \frac{C_9}{2\lambda_\phi}\alpha_k$*,* $K_r = \log(K^{1/2})$*,* $K_c \le \mathcal{O}(K^{1/4})$*, where* K *is the total number of iterations. Then, we have*

$$\frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{N}\mathbb{E}\left[\|\omega_k^i - \omega^*(\theta_k)\|^2\right] \le \mathcal{O}\left(\frac{\log^2 K}{\sqrt{K}}\right)$$

$$\frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{N}\mathbb{E}\left[\|\nabla_{\theta^i} F(\theta_k)\|^2\right] \le \mathcal{O}\left(\frac{\log^2 K}{\sqrt{K}}\right) + \mathcal{O}(\varepsilon_{app}^c + \varepsilon_{sp}), \tag{13}$$

where the constants are defined in proof.

The proof of Theorem 2 can be found in Appendix D.2. It establishes the sample complexity of $\tilde{\mathcal{O}}(\varepsilon^{-2})$ for Algorithm 2. The ε^c_{app} captures the approximation error of the critic variables, which is defined as

$$\varepsilon^c_{app} := \max_{\theta} \mathbb{E}_{s\sim\mu_\theta}\left[\left|V_{\pi_\theta}(s) - \hat{V}_{\omega^*(\theta)}(s)\right|^2\right].$$

The Algorithm 2 preserves the privacy of local actions and requires less parameters than Algorithm 1 since there is no reward estimator. The cost is that it needs to communicate $\mathcal{O}(\log(\varepsilon^{-1}))$ times for each iteration.

4.4 Proof sketch

We present the main elements for the proof of Theorem 1, which helps in understanding the difference between classical two-timescale/double-loop analysis and our single-timescale analysis. The proof of Theorem 2 follows the similar framework.

Under Markovian sampling, it is possible to show the following inequality, which characterizes the ascent of the objective.

$$\begin{aligned}\mathbb{E}[J(\theta_{k+1})] - J(\theta_k) \geq \sum_{i=1}^{N}\Big[&\frac{\alpha_k}{2}\mathbb{E}\|\nabla_{\theta^i}J(\theta_k)\|^2 + \frac{\alpha_k}{2}\mathbb{E}\|g^i_a(\xi_k,\omega^i_{k+1},\lambda^i_{k+1})\|^2 \\ &- 8C_\psi^2\alpha_k\mathbb{E}\|\omega^*(\theta_k)-\omega^i_{k+1}\|^2 - 4C_\psi^2\alpha_k\mathbb{E}\|\lambda^*(\theta_k)-\lambda^i_{k+1}\|^2\Big] \\ &- \mathcal{O}(\log^2(K)\alpha_k^2) - \mathcal{O}((\varepsilon_{app}+\varepsilon_{sp})\alpha_k). \end{aligned} \tag{14}$$

To analyze the errors of critic $\|\omega^*(\theta_k)-\omega^i_{k+1}\|^2$ and reward estimator $\|\lambda^*(\theta_k)-\lambda^i_{k+1}\|^2$, the two-timescale analysis requires $\mathcal{O}(\alpha_k) < \min\{\mathcal{O}(\beta_k),\mathcal{O}(\eta_k)\}$ in order for these two errors to converge. The double-loop approach runs lower-level update for $\mathcal{O}(\log(\varepsilon^{-1}))$ times with batch size $\mathcal{O}(\varepsilon^{-1})$ to drive these errors below ε and hence, they cannot allow inner loop size and bath size to be $\mathcal{O}(1)$ simultaneously. To obtain the convergence result for *single-timescale* update, the idea is to further upper bound these two lower-level errors by the quantity $\mathcal{O}(\alpha_k\mathbb{E}\|g^i_a(\xi_k,\omega^i_{k+1},\lambda^i_{k+1})\|^2)$ (through a series of derivations), and then eliminate these errors by the ascent term $\frac{\alpha_k}{2}\mathbb{E}\|g^i_a(\xi_k,\omega^i_{k+1},\lambda^i_{k+1})\|^2$.

We mainly focus on the analysis of critic's error through the proof sketch. The analysis for reward estimator's error follows similar procedure. We start by decomposing the error of critic as

$$\sum_{i=1}^{N}\|\omega^i_{k+1}-\omega^*(\theta_k)\|^2 = \sum_{i=1}^{N}(\|\omega^i_{k+1}-\bar{\omega}_{k+1}\|^2 + \|\bar{\omega}_{k+1}-\omega^*(\theta_k)\|^2). \tag{15}$$

The first term represents the consensus error, which can be bounded by the next lemma.

Lemma 1. *Suppose Assumptions 1 and 5 hold. Consider the sequence $\{\omega^i_k\}$ generated by Algorithm 1, then the following holds*

$$\|Q\boldsymbol{\omega}_{k+1}\| \leq \nu^{\frac{k}{K_c}-1}\|\boldsymbol{\omega}_0\| + 4\sqrt{N}C_\delta\sum_{t=0}^{k}\nu^{\frac{k-t}{K_c}-1}\beta_t,$$

where $\boldsymbol{\omega}_0 := [\omega^1_0,\cdots,\omega^N_0]^T, Q := I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$, $\nu\in(0,1)$ *is the second largest singular value of* W.

Based on Lemma 1 and follow the step size rule of Theorem 1, it is possible to show $\|Q\boldsymbol{\omega}_{k+1}\|_F^2 = \sum_{i=1}^N\|\omega^i_{k+1}-\bar{\omega}_{k+1}\|^2 = \mathcal{O}(K_c^2\beta_k^2)$. Let $K_c = \mathcal{O}(\beta_k^{-\frac{1}{2}})$, we have $\|Q\boldsymbol{\omega}_{k+1}\|_F^2 = \mathcal{O}(\beta_k)$, which maintains the optimal rate.

To analyze the second term in (15), we first construct the following Lyapunov function

$$\mathbb{V}_k := -J(\theta_k) + \|\bar{\omega}_k - \omega^*(\theta_k)\|^2 + \|\bar{\lambda}_k - \lambda^*(\theta_k)\|^2. \tag{16}$$

Then, it remains to derive an approximate descent property of the term $\|\bar{\omega}_k-\omega^*(\theta_k)\|^2$ in (16). Towards that end, our key step lies in establishing the *smoothness of the optimal critic variables* shown in the next lemma.

Lemma 2 (smoothness of optimal critic). *Suppose Assumptions 1-3 hold, under the update of Algorithm 1, there exists a positive constant* $L_{\mu,1}$ *such that for all* θ, θ', *it holds that*

$$\|\nabla\omega^*(\theta) - \nabla\omega^*(\theta')\| \leq L_{\mu,1}\|\theta - \theta'\|,$$

This smoothness property is essential for achieving our $\tilde{\mathcal{O}}(1/\sqrt{K})$ convergence rate.

To the best of our knowledge, the smoothness of $\omega^*(\theta)$ has not been justified in the literature. Equipped with Lemma 2, we are able to establish the following lemma.

Lemma 3 (Error of critic). *Under Assumptions 1-5, consider the update of Algorithm 1. Then, it holds that*

$$\begin{aligned}\mathbb{E}[\|\bar{\omega}_{k+1} - \omega^*(\theta_{k+1})\|^2] \leq\ & (1 + C_9\alpha_k)\|\bar{\omega}_{k+1} - \omega^*(\theta_k)\|^2 \\ & + \frac{\alpha_k}{4}\sum_{i=1}^{N}\|\mathbb{E}[g_a^i(\xi_k, \omega_{k+1}^i, \lambda_{k+1}^i)]\|^2 + \mathcal{O}(\alpha_k^2). \end{aligned} \tag{17}$$

$$\begin{aligned}\mathbb{E}[\|\bar{\omega}_{k+1} - \omega^*(\theta_k)\|^2] \leq\ & (1 - 2\lambda_\phi\beta_k)\|\bar{\omega}_k - \omega^*(\theta_k)\|^2 \\ & + C_{K_1}\beta_k\beta_{k-Z_K} + C_{K_2}\alpha_{k-Z_K}\beta_k. \end{aligned} \tag{18}$$

Here, $Z_K := \min\{z \in \mathbb{N}^+ | \kappa\rho^{z-1} \leq \min\{\alpha_k, \beta_k, \eta_k\}\}$, C_9, λ_ϕ *are constants specified in appendix, and* C_{K_1} *and* C_{K_2} *are of order* $\mathcal{O}(\log(K))$ *and* $\mathcal{O}(\log^2(K))$ *respectively.*

Plug (18) into (17), we can establish the approximate descent property of $\|\bar{\omega}_k - \omega^*(\theta_k)\|^2$ in (16):

$$\begin{aligned}\mathbb{E}[\|\bar{\omega}_{k+1} - \omega^*(\theta_{k+1})\|^2] \leq\ & (1 + C_9\alpha_k)(1 - 2\lambda_\phi\beta_k)\|\bar{\omega}_k - \omega^*(\theta_k)\|^2 \\ & + \frac{\alpha_k}{4}\sum_{i=1}^{N}\|\mathbb{E}[g_a^i(\xi_k, \omega_{k+1}^i, \lambda_{k+1}^i)]\|^2 \\ & + \mathcal{O}(C_{K_1}\beta_k\beta_{k-Z_K} + C_{K_2}\alpha_{k-Z_K}\beta_k). \end{aligned} \tag{19}$$

Finally, plugging (14), (17), and (19) into (16) gives the ascent of the Lyapunov function, which leads to our convergence result through steps of standard arguments.

4.5 Convergence of single-timescale decentralized NAC

The natural Actor-Critic (NAC) (Peters & Schaal, 2008) is a popular variant of AC algorithm, which enjoys the convergence to a global optimum (with compatible function approximation error) instead of a local stationary point. While our main focus is the convergence of the single-timescale AC algorithm, we find that the proof technique can be directly extended to establish the global convergence of single-timescale decentralized NAC. For reference, we design such an algorithm and provide its convergence result in Appendix E as a by-product of our single-timescale AC's analysis. To the best of our knowledge, this is the first convergence result of single-timescale NAC. However, our analysis only establishes a $\mathcal{O}(\varepsilon^{-6})$ rate for the algorithm. This result is sub-optimal compared with the existing best complexity of $\mathcal{O}(\varepsilon^{-3})$ (Chen et al., 2022), which is based on the double-loop implementation. The main reason for the sub-optimality is that in comparison with the double-loop update, the critic variables under the single-timescale update will inevitably converge slower due to the change of the actor's parameter in each iteration. Based on the classical NAC's analysis, the slower convergence of critic variables will result in a worse convergence rate of the optimality gap. Please refer to Appendix E for more discussions on the sub-optimality.

5 Numerical results

5.1 Experiment setting

We adopt the grounded communication environment proposed in (Mordatch & Abbeel, 2018). Our task consists of N agents and the corresponding N landmarks inhabited in a two-dimension world, where each

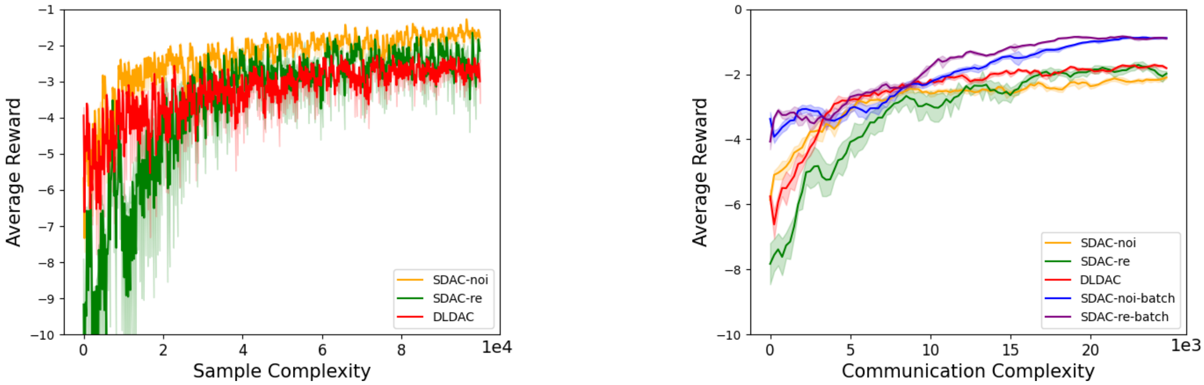

Figure 1: Averaged reward versus sample complexity and communication complexity. The vertical axis is the averaged reward over all the agents. The result is averaged over 10 Monte Carlo runs.

agent can observe the relative position of other agents and landmarks. For every discrete time step, agents take actions to move along certain directions, and receive their rewards. Agents are rewarded based on the distance to their own landmark, and penalized if they collide with other agents. The objective is to maximize the long-term averaged reward over all agents. Since we focus on decentralized setting, each agent shall not know the target landmark of others, i.e., the reward function of others. To exchange information, each agent is allowed to send their local information via a fixed communication link. Through all the experiments, the agent number N is set to be 5, and the discount factor γ is set to be 0.95.

5.2 Comparison with existing decentralized AC algorithms

In this section, we compare the proposed algorithm with existing decentralized AC algorithms under the cooperative MARL setting (Chen et al., 2022; Zeng et al., 2021) in terms of sample complexity and communication complexity. In the sequel, we refer Algorithm 1 as "SDAC-re" and Algorithm 2 as "SDAC-noi" (see Appendix 2). The algorithm proposed in (Chen et al., 2022) is referred as "DLDAC", which is based on double-loop implementation. The algorithm proposed in (Zeng et al., 2021) is denoted by "TDAC-re", which is based on two-timescale step size implementation. For comparison, we also implement a noisy reward version of "TDAC-re" and denote it by "TDAC-noi".

Comparison to double-loop decentralized AC. For "SDAC-re" and "SDAC-noi", we set $\alpha_k = 0.01(k+1)^{-0.5}$, $\beta_k = 0.1(k+1)^{-0.5}, \eta_k = 0.1(k+1)^{-0.5}, K_c = 5, \sigma = 0.5, K_r = 2$. For "DLDAC", we fix $T_c = 50$, $T_c' = 10$, $T' = 5$, $N_c = 10$, $N = 100$, $\sigma = 0.1$ [3], which is adopted by their paper (see comparisons under different hyper-parameters in Appendix A). We set $\alpha = 0.01, \beta = 0.1$ for "DLDAC" since we observe that larger step sizes will result in divergence. We have to mention that such a inner loop size $T_c = 50$ in "DLDAC" is not necessarily consistent with the theory of a double-loop algorithm, in which the loop size should be proportional to $\mathcal{O}(\varepsilon^{-1})$. The sample complexity and communication complexity results are shown in Figure 1. For the sample complexity, "SDAC-noi" enjoys a faster convergence compared with "DLDAC". In terms of communication complexity, "DLDAC" achieves better performance as it applies mini-batch technique and thereby requires less communication rounds when using the same amount of samples. Such a mini-batch approach can also be adopted to our proposed algorithms. Thus, we implement a mini-batch version of our proposed algorithms, which we refer as "SDAC-noi-batch" and "SDAC-re-batch", respectively. We set 10 as the batch size for actor, critic, and reward estimator. We can see that by applying mini-batch update, these two variants achieve significantly better communication complexity compared with "DLDAC". This is because our algorithm updates actor for more times compared with "DLDAC" under the same communication rounds.

Comparison with two-timescale decentralized AC. We fix $K_c = 1$, $K_r = 5$ for this experiment. We set $\alpha_k = 0.01(k+1)^{-0.5}$, $\beta_k = 0.1(k+1)^{-0.5}$, and $\eta_k = 0.1(k+1)^{-0.5}$ for "SDAC-re" and "SDAC-noi"; we

---

[3] Note that we adopt the notations in (Chen et al., 2022). Here, T_c is the inner loop size, T_c' is the communication number for each outer loop, T' is the communication number for reward consensus, N is the batch size for actor's update, and N_c is the batch size for critic's update.

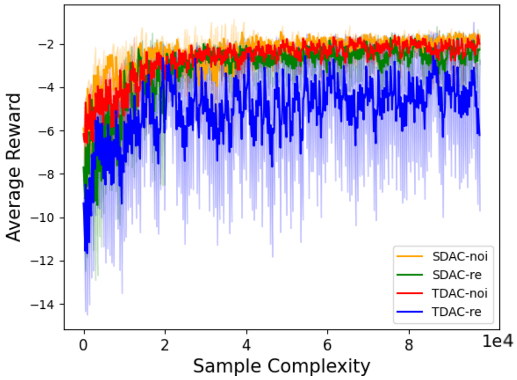

Figure 2: Comparison between the proposed algorithms and two-timescale decentralized AC algorithms (Zeng et al., 2021). The results are averaged over 10 Monte Carlo runs.

set $\alpha_k = 0.01(k+1)^{-0.6}$, $\beta_k = 0.1(k+1)^{-0.4}$, and $\eta_k = 0.1(k+1)^{-0.4}$ for "TDAC-re" and "TDAC-noi". The sample complexity is presented in Figure 2. We can observe that the convergence speed of "SDAC-noi" is slightly better than that the two-timescale counterpart "TDAC-noi". In addition, when using reward estimator for the global reward estimation, we see that "SDAC-re" has much more stable convergence behavior than "TDAC-re", and achieves significantly higher rewards.

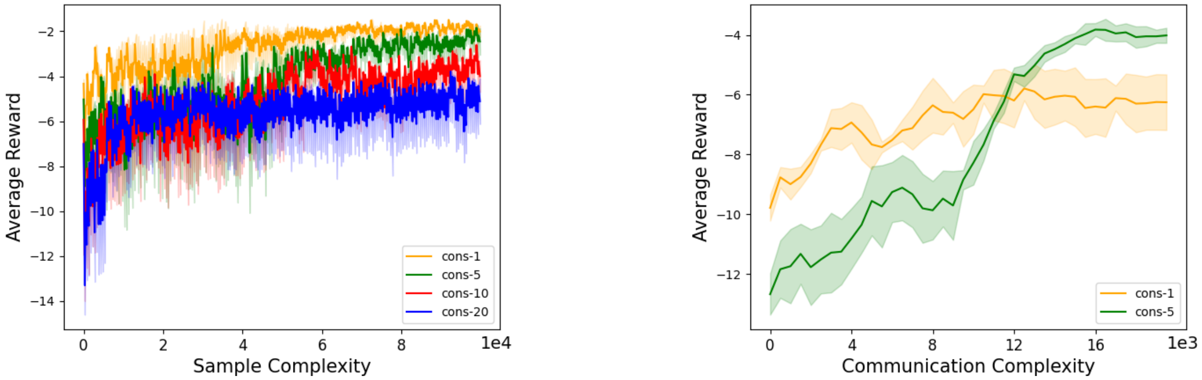

Figure 3: Ablation study on the consensus periods. The results are averaged over 10 Monte Carlo runs.

5.3 Ablation study on different choices of K_c

We compare the performance of "SDAC-noi" under different choices of consensus periods K_c. In particular, we set $\alpha_k = 0.01(k+1)^{-0.5}$, $\beta_k = 0.1(k+1)^{-0.5}$, $K_r = 2$, $\sigma = 0.5$ and examine the consensus periods K_c of 1, 5, 10, and 20, respectively. The corresponding sample complexity and communication complexity results are summarized in Figure 3. Evidently, in terms of sample complexity, the convergence becomes slower and relatively unstable as the consensus period K_c increases. Therefore, when the communication cost is low, choosing a small K_c will yield a better performance. We also plot the communication complexity under the consensus periods of 1 and 5. We can see that the communication complexity of "cons-5" outperforms "cons-1" after 12×10^3 communications. Thus, when the communication cost is expensive and high averaged reward is required, one may use large K_c and run the algorithm for a relatively large number of iterations.

6 Conclusion and future direction

In this paper, we studied the convergence of fully decentralized AC algorithm under practical single-timescale update for the first time. We designed such an algorithm which maintains the optimal sample complexity of

$\tilde{\mathcal{O}}(\varepsilon^{-2})$ under less communications. We also proposed a variant to preserve the privacy of local actions by communicating noisy rewards. Extensive simulation results demonstrate the superiority of our algorithms' empirical performance over existing decentralized AC algorithms. However, directly extending our single-timescale AC's analysis technique to single-timescale NAC will result in a sub-optimal sample complexity. We leave the study on improving the convergence rate and design a more efficient single-timescale NAC algorithm as promising future directions.

References

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *ArXiv:1908.00261*, 2019.

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in markov decision processes. In *Conference on Learning Theory (COLT)*, pp. 64–66, 2020.

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

Jalaj Bhandari, Daniel Russo, and Raghav Singal. A finite time analysis of temporal difference learning with linear function approximation. In *Conference on Learning Theory (COLT)*, pp. 1691–1692, 2018.

Tianyi Chen, Yuejiao Sun, and Wotao Yin. Closing the gap: Tighter analysis of alternating stochastic gradient methods for bilevel problems. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021a. URL `https://openreview.net/forum?id=OItvP2-i9j`.

Ziyi Chen, Yi Zhou, and Rongrong Chen. Multi-agent off-policy td learning: Finite-time analysis with near-optimal sample complexity and communication complexity. *arXiv preprint arXiv:2103.13147*, 2021b.

Ziyi Chen, Yi Zhou, Rong-Rong Chen, and Shaofeng Zou. Sample and communication-efficient decentralized actor-critic algorithms with finite-time analysis. In *International Conference on Machine Learning*, pp. 3794–3834. PMLR, 2022.

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

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International Conference on Machine Learning*, pp. 1407–1416. PMLR, 2018.

Zuyue Fu, Zhuoran Yang, and Zhaoran Wang. Single-timescale actor-critic provably finds globally optimal policy. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=pqZV_srUVmK`.

Hongyi Guo, Zuyue Fu, Zhuoran Yang, and Zhaoran Wang. Decentralized single-timescale actor-critic on zero-sum two-player stochastic games. In *International Conference on Machine Learning*, pp. 3899–3909. PMLR, 2021.

FNU Hairi, Jia Liu, and Songtao Lu. Finite-time convergence and sample complexity of multi-agent actor-critic reinforcement learning with average reward. In *International Conference on Learning Representations*, 2022.

Sham M Kakade. A natural policy gradient. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pp. 1531–1538, 2002.

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

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Yixuan Lin, Kaiqing Zhang, Zhuoran Yang, Zhaoran Wang, Tamer Başar, Romeil Sandhu, and Ji Liu. A communication-efficient multi-agent actor-critic algorithm for distributed reinforcement learning. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 5562–5567, 2019.

Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pp. 157–163. Elsevier, 1994.

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

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Alex Olshevsky and Bahman Gharesifard. A small gain analysis of single timescale actor critic. *arXiv preprint arXiv:2203.02591*, 2022.

Shayegan Omidshafiei, Jason Pazis, Christopher Amato, Jonathan P How, and John Vian. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *International Conference on Machine Learning*, pp. 2681–2690. PMLR, 2017.

Jan Peters and Stefan Schaal. Natural actor-critic. *Neurocomputing*, 71(7-9):1180–1190, 2008.

Shuang Qiu, Zhuoran Yang, Jieping Ye, and Zhaoran Wang. On the finite-time convergence of actor-critic algorithm. In *Optimization Foundations for Reinforcement Learning Workshop at Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 4295–4304. PMLR, 2018.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Han Shen, Kaiqing Zhang, Mingyi Hong, and Tianyi Chen. Asynchronous advantage actor critic: Non-asymptotic analysis and linear speedup. *ArXiv:2012.15511*, 2020.

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

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 5887–5896. PMLR, 2019.

Jun Sun, Gang Wang, Georgios B Giannakis, Qinmin Yang, and Zaiyue Yang. Finite-sample analysis of decentralized temporal-difference learning with linear function approximation. In *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 4485–4495, 2020.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pp. 1057–1063, 2000.

John N Tsitsiklis and Benjamin Van Roy. Analysis of temporal-diffference learning with function approximation. In *Advances in neural information processing systems (NIPS)*, pp. 1075–1081, 1997.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

Yue Wang, Shaofeng Zou, and Yi Zhou. Non-asymptotic analysis for two time-scale TDC with general smooth function approximation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 9747–9758. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper/2021/file/50e207ab6946b5d78b377ae0144b9e07-Paper.pdf`.

Yue Frank Wu, Weitong Zhang, Pan Xu, and Quanquan Gu. A finite-time analysis of two time-scale actor-critic methods. *Advances in Neural Information Processing Systems*, 33:17617–17628, 2020.

Pan Xu, Felicia Gao, and Quanquan Gu. An improved convergence analysis of stochastic variance-reduced policy gradient. In *Proc. International Conference on Uncertainty in Artificial Intelligence (UAI)*, 2019.

Tengyu Xu and Yingbin Liang. Sample complexity bounds for two timescale value-based reinforcement learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, pp. 811–819. PMLR, 2021.

Tengyu Xu, Zhe Wang, and Yingbin Liang. Improving sample complexity bounds for (natural) actor-critic algorithms. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

Tengyu Xu, Zhuoran Yang, Zhaoran Wang, and Yingbin Liang. Doubly robust off-policy actor-critic: Convergence and optimality. *ArXiv:2102.11866*, 2021.

Chao Yu, Xin Wang, Xin Xu, Minjie Zhang, Hongwei Ge, Jiankang Ren, Liang Sun, Bingcai Chen, and Guozhen Tan. Distributed multiagent coordinated learning for autonomous driving in highways based on dynamic coordination graphs. *IEEE Transactions on Intelligent Transportation Systems*, 21(2):735–748, 2019.

Siliang Zeng, Tianyi Chen, Alfredo Garcia, and Mingyi Hong. Learning to coordinate in multi-agent systems: A coordinated actor-critic algorithm and finite-time guarantees. *arXiv preprint arXiv:2110.05597*, 2021.

Haifeng Zhang, Weizhe Chen, Zeren Huang, Minne Li, Yaodong Yang, Weinan Zhang, and Jun Wang. Bi-level actor-critic for multi-agent coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7325–7332, 2020.

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

Kaiqing Zhang, Alec Koppel, Hao Zhu, and Tamer Başar. Global convergence of policy gradient methods to (almost) locally optimal policies. *arXiv preprint arXiv:1906.08383*, 2019.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, pp. 321–384, 2021.