2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Federated TD Learning in Heterogeneous Environments with Average Rewards: A Two-timescale Approach with Polyak-Ruppert Averaging

Ankur Naskar[a], Gugan Thoppe[a], Abbasali Koochakzadeh[b], and Vijay Gupta[b]

Abstract— **Federated Reinforcement Learning (FRL) provides a promising way to speedup training in reinforcement learning using multiple edge devices that can operate in parallel. Recently, it has been shown that even when these edge devices have access to different dynamic models, an optimal convergence rate that has a linear speedup proportional to the number of devices is achievable. However, this result requires that the stepsize in the algorithm be chosen in a manner dependent on the unknown model parameters. Also, it applies only to a discounted setting, which has been argued to fit episodic tasks better than continuing control tasks. In this paper, we obtain finite-time bounds for heterogeneous FRL with average rewards. We show that the optimal convergence rate with a linear speedup is possible even with a universal stepsize choice, independent of the underlying dynamics. To achieve our result, we modify the existing one-timescale FRL method to a novel two-timescale variant that additionally incorporates iterate averaging.**

I. INTRODUCTION

With the proliferation of edge devices such as scanners, smartphones, medical devices, scientific instruments, and autonomous vehicles, there has been a massive growth in the amount of data that could conceivably be used for training Machine Learning (ML) models. However, transmitting this data from the edge devices to a single centralized location can easily overwhelm the network bandwidth and also raise privacy concerns. The paradigm of Federated Learning (FL) has evolved to address these concerns by allowing model training to be done primarily by the edge devices themselves [1], [2]. Fundamentally, FL consists of three steps that are repeated in a cyclic manner. First, the edge devices locally train an ML model. Next, the server gathers these models, possibly by sampling the devices, and then aggregates them. Finally, the server transmits this global model to the edge devices. The core challenge of FL is that the edge devices and their contributions may be heterogeneous. The most obvious form of this heterogeneity is that of data heterogeneity, meaning that the edge devices operate in distinct environments and collect data from distributions that are not identical to each other. Much of the literature in FL accordingly provides ways to combat such heterogeneity.

A branch of ML that has seen a surge of interest recently is Reinforcement Learning (RL) [3], [4], [5], [6], [7], wherein an agent is required to learn an optimal strategy (or a policy) for a control system. This system's dynamics is modeled as a Markov Decision Process (MDP) so that its future state (or its probability distribution) depends only on its current state and an action performed by an agent. After each action, the agent receives a scalar feedback, called the reward, that quantifies the immediate gains under that action. The overall goal in RL is to find a policy for the agent that optimizes a suitable cumulative sum of these instantaneous rewards. There are two main ways in which the above cumulative sum (of rewards) has been defined in the literature. One way is to look at a discounted sum of the instantaneous rewards (with some discount factor $\gamma \in [0, 1)$) over some time horizon. However, it has been argued that such discounting is suitable for episodic tasks but leads to high near-term performance rather than to high long-term performance in continuing or infinite-horizon control tasks [8], [9], [10]. The alternative is the average-reward setting, which is our focus.

Some recent works have merged the FL and RL paradigms and studied Federated Reinforcement Learning (FRL) [11], [12], [13], [14]. One advantage of FRL is that a 'divide and conquer' approach to generating the usually enormous amount of data required to explore the different aspects of the environment becomes conceivable. Moreover, the different edge devices can now perform computations in parallel, leading to a possible speedup in terms of the number of participating devices. Initial works show that this intuition is true, at least in the case when each node has access to the same model for the system process, e.g., [15].

However, as stated above, heterogeneity is a key concern in FL. In FRL, this may manifest as different edge devices working with (slightly) different dynamic models. For instance, to design a controller for an autonomous car, we can try to utilize data from several cars; however, every car operates in a different environment with different configurations. It is important to note that, due to the global aggregation step in FL, the designed controller may end up not being optimal for the dynamic model at any single edge device. Hence, one typically works with an alternative goal in heterogenous FRL, which is to find a 'universal' controller that performs well across all the edge models. An intriguing question is whether the speedup from the homogeneous case can be achieved in the heterogeneous case as well.

Some recent works [16], [17] have studied this question.

The work was partially supported by Purdue University and the Science and Engineering Research Board (SERB) of India through the Overseas Visiting Doctoral Fellowship, ARO under grant W911NF2310266 and ONR under grant N000142312604. It was also supported in part by DST-SERB's Core Research Grant CRG/2021/008330, the Indo-French Centre for the Promotion of Advanced Research—CEFIPRA (7102-1), the Walmart Centre for Tech Excellence, and the Pratiksha Trust Young Investigator Award.

[a]A. Naskar (ankurnaskar@iisc.ac.in) and G. Thoppe (gthoppe@iisc.ac.in) are with the Dept. of Computer Science and Automation, Indian Institute of Science, Bengaluru, India

[b]A. Koochakzaadeh (akoochak@purdue.edu) and V. Gupta (gupta869@purdue.edu) are with the Dept. of Electrical and Computer Engineering, Purdue University, IN, USA

Specifically, [17] considered policy evaluation with linear function approximation using TD learning and proved that the optimal convergence rate and a linear speedup persists even with heterogeneity. However, their setup considers discounted rewards, which as stated above, is less suitable for continuing control tasks than average rewards. Moreover, for the optimal rate, the algorithm in [17] is required to choose stepsizes in a manner that depend on the unknown dynamic models at the edge devices, which is practically infeasible.

In this paper, we consider heterogeneous FRL with an average-reward criterion and show that the optimal rate with a linear speedup is still possible by choosing stepsizes that do not depend on the unknown dynamics at the edge devices. A formal summary of our key contributions is as follows. We propose a novel *two-timescale* variant of TD learning for policy evaluation in heterogeneous settings with the *average-reward* criterion and linear function approximation. In this variant, we adapt iterate averaging from single-agent RL [18], [19], [20] to a federated setup with heterogeneous MDPs. Our main result shows that our algorithm achieves the *optimal rate* with a *linear speedup* in sample complexity in terms of the number of agents. Importantly, we show that this optimality occurs for a *universal* stepsize choice that is independent of the underlying dynamics.

The rest of the paper is organized as follows. We begin with formulating the problem in Section II. Our proposed algorithm and our main result (Theorem 3.1) are given in Section III. We prove our main result in Section IV and conclude with some open directions in Section VI.

Algorithm 1: Our proposed AvgFedTD(0) algorithm

Input : Policy μ, step-size sequence (β_t), feature vectors $\{\phi(s) : s \in \mathcal{S}\}$, initial average reward estimate $r_0 \in \mathbb{R}$, and initial global model parameter $\theta_0 \in \mathbb{R}^d$.

Initialize: $\bar{\theta}_0 = \theta_0$ and $r_0^i = r_0, \forall i \in [N]$.

for *each iteration* $t = 0, 1, \ldots, T-1$:

Each agent $i \in [N]$ **in parallel**

Observe $(s_t^i, a_t^i, \hat{s}_t^i)$, where $s_t^i \sim d_i^\mu$, $a_t^i \sim \mu(\cdot|s_t^i)$, and $\hat{s}_t^i \sim \mathcal{P}_i(\cdot|s_t^i, a_t^i)$.

Compute local TD error
$$\delta_{t+1}^i = [\mathcal{R}_i(s_t^i, a_t^i) - r_t]\phi(s_t^i) + \phi(s_t^i)[\phi^\top(\hat{s}_t^i) - \phi^\top(s_t^i)]\theta_t.$$

Update local average reward estimate
$$r_{t+1}^i = r_t^i + \tfrac{1}{t+1}[\mathcal{R}_i(s_t^i, a_t^i) - r_t^i].$$

Send $(\delta_{t+1}^i, r_{t+1}^i)$ to central server.

Central server

Update global model parameter
$$\theta_{t+1} = \theta_t + \tfrac{\beta_t}{N}\textstyle\sum_{i\in[N]} \delta_{t+1}^i.$$

Update Polyak-Ruppert average
$$\bar{\theta}_{t+1} = \bar{\theta}_t + \tfrac{1}{t+1}[\theta_t - \bar{\theta}_t].$$

Update average reward estimate
$$r_{t+1} = \tfrac{1}{N}\textstyle\sum_{i\in[N]} r_{t+1}^i.$$

Send (θ_{t+1}, r_{t+1}) to each agent $i \in [N]$.

end

II. SETUP AND PROBLEM FORMULATION

Our framework comprises N agents (also referred to as clients or nodes), with the i-th agent having access to an MDP $\mathcal{M}_i := (\mathcal{S}, \mathcal{A}, \mathcal{R}_i, \mathcal{P}_i)$. Here, $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, respectively, and are assumed to be finite and common among all the MDPs. Further, $\mathcal{R}_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and $\mathcal{P}_i : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ are the local reward and probability transition functions at agent $i \in [N]$ and can potentially vary from one agent to the other. The notation $\Delta(\mathcal{S})$ stands for the set of distributions on $\mathcal{S}$, and $[N] := \{1, \ldots, N\}$. Our problem is to design an FRL algorithm that leverages the above setup to estimate the value function of a stationary policy $\mu : \mathcal{S} \to \Delta(\mathcal{A})$. In particular, our algorithm's output should approximate μ's value function in the column space of a given feature matrix $\Phi \in \mathbb{R}^{|\mathcal{S}|\times d}$ for some $1 \le d \ll |\mathcal{S}|$.

Under the average-reward criterion, μ's value function with respect to the MDP $\mathcal{M}_i$ can be measured using two notions. First, the average reward $r_i^\mu \in \mathbb{R}^{|\mathcal{S}|}$ is given by

$$r_i^\mu(s) := \liminf_{T\to\infty} \frac{1}{T}\mathbb{E}\Big[\sum_{t=0}^{T-1} \mathcal{R}_i(s_t, a_t)|s_0 = s\Big], \quad s \in \mathcal{S}, \tag{1}$$

where the expectation is with respect to the distribution of the state-action trajectory $s_0, a_0, \ldots, s_{T-1}, a_{T-1}$, in which $a_t \sim \mu(\cdot|s_t)$ and $s_{t+1} \sim \mathcal{P}_i(\cdot|s_t, a_t)$. In contrast, the differential value function V_i^μ is the fixed point of the differential Bellman operator $T_i^\mu : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$ which is given by

$$T_i^\mu V = \mathcal{R}_i^\mu - r_i^\mu + \mathcal{P}_i^\mu V. \tag{2}$$

Above, $\mathcal{R}_i^\mu(s) = \sum_{a\in\mathcal{A}} \mu(a|s)\mathcal{R}_i(s, a)$ and $\mathcal{P}_i^\mu(s, s') \equiv \mathcal{P}_i^\mu(s'|s) = \sum_{a\in\mathcal{A}} \mu(a|s)\mathcal{P}_i(s'|s, a)$. Throughout this work, we presume that the data (i.e, the states and actions) observed at different agents are independent. Additionally, we make the following standard assumption [10], [17]:

$\mathcal{A}_1$) **Ergodicity**: For any $i \in [N]$, the Markov chain $(\mathcal{S}, \mathcal{P}_i^\mu)$ induced by the policy μ is irreducible and aperiodic.

This assumption guarantees that the Markov chain $(\mathcal{S}, \mathcal{P}_i^\mu)$ has a unique and positive stationary distribution d_i^μ; further, this Markov chain is ergodic, and, for each $s \in \mathcal{S}$, we have $r_i^\mu(s) = (d_i^\mu)^\top \mathcal{R}_i^\mu =: r_i^*$. Note that r_i^* is independent of s.

In the above notations, our FRL algorithm's goals can be restated as follows. It should output a vector $\theta \in \mathbb{R}^d$ and a scalar $r \in \mathbb{R}$ such that $\Phi\theta$ (resp. r) is simultaneously close to V_i^μ (resp. r_i^*) for each i. Moreover, our algorithm's convergence rate should be optimal and its iteration complexity—i.e., the number of iterations needed to find such a θ and r—should decrease linearly with the number N of agents. Finally, the stepsize choice to achieve this optimal rate should not depend on unknown problem-dependent constants.

III. PROPOSED ALGORITHM AND MAIN RESULT

Our novel algorithm for policy evaluation in the heterogeneous FL setup under the average reward criterion is given

in Algorithm 1. There are two phases in each iteration of this algorithm. In the first phase, all agents work in parallel to compute their local TD errors and their local estimates of the average reward, i.e., the δ_t^i's and the r_t^i's, which are then shared with a central server. In the second phase, the server aggregates these quantities to obtain global estimates for the differential value function and the average reward, which are then broadcast to all the agents. To compute the local TD error δ_t^i, we assume that agent i has access to the current state $s_t^i \sim d_i^\mu$ of its local MDP $\mathcal{M}_i$, and can take action $a_t^i \sim \mu(s_t^i)$, after which it gets to see the subsequent local state $\hat{s}_t^i \sim \mathcal{P}_i^\mu(\cdot|s_t^i, a_t^i)$, and the local instantaneous reward $\mathcal{R}_i(s_t^i, a_t^i)$. We assume that the tuple $(s_t^i, a_t^i, \hat{s}_t^i)$ is independent across iterations and also agents. Note that the information about the local states, actions, and rewards is not shared directly with the server, following FL principles.

We now discuss how our algorithm differs from [17, Algorithm 1] and also the latter's limitation. First, note that the algorithm in [17] addresses the exponentially discounted case, whereas ours focuses on the average-reward scenario. Hence, to enable comparison, we first naively extend the algorithm in [17] to the average-reward setting by incorporating ideas from [10, Algorithm1], which is tailored for policy evaluation in the *single-agent* and *average-reward* context. The resulting algorithm closely mirrors our Algorithm 1, with the following crucial modifications: (i) in Step 4, the $1/(t+1)$ stepsize is replaced by $c_\alpha \beta_t$; and (ii) Step 7 is omitted.

Regarding limitation, it can be guessed from [10, Theorem 1] and [17, Theorem 2] that the optimal convergence rates for θ_t and r_t in this naive extension—which will have a linear speedup in terms of sample complexity—is guaranteed only when the stepsize β_t is of the form $c_1/(c_2+t)$. Moreover, the constants c_1, c_2, and c_α need to be such that $2 < c_1\Delta < 2c_2$ and $c_\alpha > \Delta + \sqrt{\Delta^{-2} - 1}$, where $\Delta := \min_{\theta \in \mathbb{R}^d \setminus \{0\}} \theta^\top A \theta$ and $A = \frac{1}{N}\sum_{i\in[N]} A_i$ with $A_i := \Phi^\top D_i^\mu (I - \mathcal{P}_i^\mu)\Phi$ and $D_i^\mu := \text{diag}(d_i^\mu)$ for all $i \in [N]$. Clearly, this stepsize choice is impractical, as it necessitates the knowledge of Δ, which depends on $\mathcal{P}_i^\mu$, an a priori unknown matrix.

Our approach differs from the aforementioned naive algorithm in the following two crucial ways. The first is that we have the additional Step 7, wherein we perform a *Polyak-Ruppert* [19], [18] running averaging of the global θ-estimates, i.e., we compute $\bar{\theta}_t = \frac{1}{t}\sum_{k=0}^{t-1} \theta_k$ at each time instance t. The second is in changing the updating approach of θ_t and r_t^i from one-timescale to *two-timescales*. Put differently, in the naive extension, both θ_t and r_t^i are updated using stepsizes that differ only by a constant factor. In contrast, in our algorithm, we update these quantities using the stepsizes β_t and $1/(t+1)$, respectively, and we assume that their ratio goes to ∞ as $t \to \infty$. As an example, this condition holds for $\beta_t = 1/(t+1)^\beta$ when β is some constant in $(0, 1)$.

Because of the above two modifications, we now show that our algorithm does not have the impracticality issue inherent in the above naive algorithm. That is, it achieves the optimal convergence rate and a linear speedup in the associated sample complexity without requiring the stepsize β_t to depend on unknown problem-dependent parameters.

We need the following assumptions to formally state our main result. Let $\|\cdot\|$ be the Euclidean norm.

- $\mathcal{A}_\mathbf{2}$) **Heterogeneity bound**: $\exists\ \epsilon_p \geq 0$ and $\epsilon_r \geq 0$ such that $|\mathcal{R}_i(s,a) - \mathcal{R}_j(s,a)| \leq \epsilon_r$ and
$$|\mathcal{P}_i(s'|s,a) - \mathcal{P}_j(s'|s,a)| \leq \epsilon_p\ \mathcal{P}_i(s'|s,a)$$
$\forall i, j \in [N]$, $s \neq s' \in \mathcal{S}$, and $a \in \mathcal{A}$.
- $\mathcal{A}_\mathbf{3}$) **Bounded rewards**: $\exists R_{\max} > 0$ such that $|\mathcal{R}_i(s,a)| \leq R_{\max}\ \forall i \in [N], \forall s \in \mathcal{S}$, and $\forall a \in \mathcal{A}$.
- $\mathcal{A}_\mathbf{4}$) **Conditions on the feature matrix**: The matrix Φ has full-column rank with $\|\Phi\| \leq 1$. Additionally, the column space of Φ does not contain the vector of all ones, i.e., $\mathbb{1} \notin \{\Phi\theta : \theta \in \mathbb{R}^d\}$.

Assumption $\mathcal{A}_\mathbf{2}$ puts an upper bound on the heterogeneity among the local environments. Separately, Assumption $\mathcal{A}_\mathbf{4}$, which is borrowed from [10], along with $\mathcal{A}_\mathbf{1}$ is required for the positive definiteness of each A_i [10, Lemma 2]. The positive definiteness of A follows from that of the A_i's.

We also need a few notations. For all $i \in [N]$, let $b_i := \Phi^\top D_i^\mu \mathcal{R}_i^\mu$, $v_i := \Phi^\top D_i^\mu \mathbb{1}$, and $\theta_i^* := A_i^{-1}(b_i - v_i r_i^*)$. Separately, let $b := \frac{1}{N}\sum_{i\in[N]} b_i$, $v := \frac{1}{N}\sum_{i\in[N]} v_i$, and $r^* := \frac{1}{N}\sum_{i\in[N]} r_i^*$. Finally, let $\theta^* := A^{-1}(b - vr^*)$.

We are now ready to state our main result on the finite-time convergence of our AvgFedTD(0) algorithm.

Theorem 3.1: Assume conditions $\mathcal{A}_\mathbf{1}$—$\mathcal{A}_\mathbf{4}$. Let $(\bar{\theta}_t, r_t)$ be the iterates generated by Algorithm 1 with $\beta_t = \frac{1}{(t+1)^\beta}$ for a $\beta \in (0,1)$. Then, $\forall i \in [N]$ and $T > 0$,
$$\mathbb{E}|r_T - r_i^*|^2 \leq \frac{C_1}{(T+1)^2} + \frac{C_2}{N(T+1)} + H_r(\epsilon_p, \epsilon_r)$$
and
$$\mathbb{E}\|\bar{\theta}_T - \theta_i^*\|^2 \leq \frac{C_3 \ln(T)}{(T+1)^2} + \frac{C_4}{N(T+1)} + H_\theta(\epsilon_p, \epsilon_r),$$
where C_1—C_4 are some non-negative constants, while $H_r(\epsilon_p, \epsilon_r)$ and $H_\theta(\epsilon_p, \epsilon_r)$ are special constants called heterogeneity gaps and are given by the expressions in Table I.

Remark 3.2: Our result shows that the expected squared error in our estimates for θ_i^* and r_i^* is $O\left(\frac{1}{NT}\right)$ at time T, modulo the constant heterogeneity gaps H_r and H_θ. These gaps decay to 0 as $\epsilon_p, \epsilon_r \to 0$. In the single-agent setting, our algorithm would have converged to θ_1^* at the optimal rate of $O\left(\frac{1}{T}\right)$. In contrast, when $N > 1$, our algorithm guarantees convergence only to a H_θ-sized neighborhood of θ_1^* (or alternatively of any θ_i^*), but with the optimal rate and a sample complexity speedup proportional to N. Thus, collaborating in a heterogeneous setup leads to a speedup at the cost of accuracy, mirroring the conclusion in [17].

Remark 3.3: The most significant part of our result is that we obtain the $O\left(\frac{1}{NT}\right)$ rate without requiring our stepsize β_t to depend on unknown problem-specific constants. This contrasts the results in [10] and [17] where, as discussed at the beginning of this section, the optimal rate is guaranteed only if $\beta_t = c_1/(c_2+t)$ and c_1 and c_2 satisfy some constraints depending on the unknown A, $A_1, \ldots, A_N$ matrices. Thus, our results hold for practically realizable stepsizes, while the ones in [10] and [17] do not.

TABLE I: Table of constants.

Constants	Values
$C_d(\epsilon_p)$	$\left(\frac{1+\epsilon_p}{1-\epsilon_p}\right)^{\lvert\mathcal{S}\rvert} - 1 = 2\lvert\mathcal{S}\rvert\epsilon_p + O(\epsilon_p^2)$
$C_A(\epsilon_p)$	$\epsilon_p\sqrt{\lvert\mathcal{S}\rvert} + C_d(\epsilon_p)\ (1+\sqrt{\lvert\mathcal{S}\rvert})$
$C_b(\epsilon_p, \epsilon_r)$	$\sqrt{\lvert\mathcal{S}\rvert}[2\epsilon_r + 3C_d(\epsilon_p)R_{\max}]$
$H_r(\epsilon_p, \epsilon_r)$	$2\lvert\mathcal{S}\rvert\left[\epsilon_r^2 + R_{\max}^2 C_d^2(\epsilon_p)\right]$
$H_\theta(\epsilon_p, \epsilon_r)$	$\max_{i\in[N]} \frac{2\kappa^2(A_i)\ \lVert A_i\rVert^2\lVert\theta_i^*\rVert^2}{[\lVert A_i\rVert - \kappa(A_i)C_A(\epsilon_p)]^2}\left[\frac{C_A^2(\epsilon_p)}{\lVert A_i\rVert^2} + \frac{C_b^2(\epsilon_p,\epsilon_r)}{\lVert b_i - v_i r_i^*\rVert^2}\right]$ where $\kappa(A_i) := \sigma_{\max}(A_i)/\sigma_{\min}(A_i)$

Remark 3.4: While multi-timescale approaches have been used and analyzed previously [21], [22], [23], [24], [25], their motivation broadly was either to simplify the analysis by decoupling the behavior of two or more iterates, e.g., [26], or to remove the correlation between the quantities estimated using the same set of random observations, e.g., [27]. However, the algorithms in [26] and [27] run well even in the one-timescale setting, showing that the use of two-timescales was not strictly required. In contrast, our work shows that the two-timescale approach leads to a practically-realizable stepsize for optimal speedup, while the single-timescale-based one does not.

IV. PROOF SKETCH OF MAIN RESULT

Due to space constraints, we provide only a sketch of the arguments that we use for proving Theorem 3.1.

We begin by rewriting our update rules in Algorithm 1 in a form that enables our analysis. For each $i \in [N]$, let

$$W_{t+1}^{(i)} := \mathcal{R}_i(s_t^i, a_t^i) - (d_i^\mu)^\top \mathcal{R}_i^\mu \tag{3}$$

$$M_{t+1}^{(i)} := [\mathcal{R}_i(s_t^i, a_t^i)\phi(s_t^i) - b_i] - [\phi(s_t^i) - v_i]r_t - [\phi(s_t^i)(\phi^\top(s_t^i) - \phi^\top(\hat{s}_t^i)) - A_i]\theta_t, \tag{4}$$

where A_i, b_i, and v_i are as in Section III, while d_i^μ and $\mathcal{R}_i^\mu$ are as in Section II. Further, $\forall t \geq 0$, let

$$\rho_t := r_t - r^*, \quad \Delta_t := \theta_t - \theta^*, \quad \text{and} \quad \bar{\Delta}_t := \bar{\theta}_t - \theta^*. \tag{5}$$

Then, we get the following alternative update rules:

$$\rho_{t+1} = \big(1 - \frac{1}{t+1}\big)\rho_t + \frac{1}{t+1}W_{t+1} \tag{6}$$

$$\Delta_{t+1} = (I - \beta_t A)\Delta_t - \beta_t v\rho_t + \beta_t M_{t+1} \tag{7}$$

$$\bar{\Delta}_{t+1} = \bar{\Delta}_t + \frac{1}{t+1}[\Delta_t - \bar{\Delta}_t], \tag{8}$$

where

$$W_{t+1} := \frac{1}{N}\sum_{i=1}^N W_{t+1}^{(i)} \quad \text{and} \quad M_{t+1} := \frac{1}{N}\sum_{i=1}^N M_{t+1}^{(i)}.$$

Clearly, both (W_{t+1}) and (M_{t+1}) are Martingale-difference sequences w.r.t. the filtration $(\mathcal{F}_t)$, where $\mathcal{F}_t$ is the σ-field $\sigma(\theta_0, r_0, s_k^i, a_k^i, \hat{s}_k^i : i \in [N], 0 \leq k \leq t-1)$.

Next, we provide an explanation for the non-decaying constants $H_r(\epsilon_p, \epsilon_r)$ and $H_\theta(\epsilon_p, \epsilon_r)$ in Theorem 3.1, which we refer to as the heterogeneity gaps. Applying stochastic approximation theory [28, Chapter 2] to (6), the only potential point that r_t can converge to is r^*. Similarly, from (7) and (8), the only point where Δ_t and, hence, $\bar{\Delta}_t$ can converge to is θ^*. In contrast, the standard single-agent TD(0) [10], if run by any agent i without communication, would generate a sequence (r_t, θ_t) that would converge to (r_i^*, θ_i^*). This difference is due to the heterogeneity in our FL setup and is the reason behind $H_r(\epsilon_p, \epsilon_r)$ and $H_\theta(\epsilon_p, \epsilon_r)$ in Theorem 3.1. Our first lemma shows that H_θ (resp. H_r) bounds the gap between θ^* (resp. r^*) and θ_i^* (resp. r_i^*).

Lemma 4.1: For each $i \in [N]$,

$$|r^* - r_i^*| \leq \sqrt{H_r(\epsilon_p, \epsilon_r)}$$
$$\|\theta^* - \theta_i^*\| \leq \sqrt{H_\theta(\epsilon_p, \epsilon_r)},$$

where $H_r(\epsilon_p, \epsilon_r)$ and $H_\theta(\epsilon_p, \epsilon_r)$ are as defined in Table I.

Remark 4.2: The constants $H_r(\epsilon_p, \epsilon_r) = H_\theta(\epsilon_p, \epsilon_r) \to 0$ when the heterogeneity parameters ϵ_p and ϵ_r decay to 0. This can be checked from Table I.

The proof mirrors, mutatis mutandis, the one used to derive [17, Theorem 1], which provides a similar result for the discounted case. Note that the TD(0) algorithm in the discounted case does not involve (r_t) updates; hence, Theorem 1 in ibid does not have the H_r heterogenity gap.

The next two lemmas quantify the convergence rates of r_t and θ_t to r^* and θ^*, respectively.

Lemma 4.3: For $T \geq 1$, $\mathbb{E}\rho_T^2 \leq \frac{4R_{\max}^2}{NT} = O\left(\frac{1}{NT}\right)$.

Remark 4.4: Note that $\mathbb{E}\rho_T^2$ does not depend on ρ_0. This is not surprising since, from (6), we have $\rho_1 = W_1$ which clearly does not depend on ρ_0.

The proof follows by showing that for any $t \geq 0$,

$$\mathbb{E}\rho_{t+1}^2 = \left(\frac{t}{t+1}\right)^2 \mathbb{E}\rho_t^2 + \frac{1}{N^2(t+1)^2}\sum_{i=1}^N \mathbb{E}(W_{t+1}^i)^2,$$

which using Assumption $\mathcal{A}_3$ yields

$$\mathbb{E}\rho_{t+1}^2 \leq \left(\frac{t}{t+1}\right)^2 \mathbb{E}\rho_t^2 + \frac{4R_{\max}^2}{N(t+1)^2}.$$

A simple inductive argument now gives the desired result.

Lemma 4.5: For $t \geq 1$, we have $\mathbb{E}\|\Delta_t\|^2 = O\left(\frac{\beta_t}{N}\right)$.

The proof of this result mirrors the one used to derive [29, Theorem 3.1], which looks at similar bounds for the discounted case in the single-agent settting.

Next, we bound $\mathbb{E}\|\bar{\Delta}_T\|^2$ which concerns the error in the iterate average $\bar{\theta}_T$.

Lemma 4.6: There exists constants $K, C_M \geq 0$ such that

$$\mathbb{E}\|\bar{\Delta}_t\|^2 \leq \frac{3K^2\mathbb{E}\|\Delta_0\|^2}{t^2} + \frac{6|\mathcal{S}|K^2}{t^2}\mathbb{E}\rho_0^2 + \frac{24|\mathcal{S}|K^2R_{\max}^2}{Nt} + \frac{3C_M}{Nt^2}\sum_{k=0}^{t-2}[1 + \mathbb{E}\|\Delta_k\|^2 + \mathbb{E}\rho_k^2] \tag{9}$$

for any $t \geq 1$.

To derive the above result, we make use of multiple steps. First, we use [18, Lemma 2] to show that if (Δ_t) follows

the linear stochastic approximation rule (7), then its Polyak-Ruppert average ($\bar{\Delta}_t$) satisfies

$$\bar{\Delta}_t = \frac{\alpha_0^t \Delta_0}{t\beta_0} - \frac{1}{t}\sum_{k=0}^{t-2} \alpha_k^{t-1} v\rho_k + \frac{1}{t}\sum_{k=0}^{t-2} \alpha_k^{t-1} M_{k+1}, \quad (10)$$

where

$$\alpha_k^\ell = \beta_k \sum_{i=k}^{\ell-1} \prod_{j=k}^{i-1} (I - \beta_j A).$$

Additionally, using Lemma 2 in ibid, we get that there exists $K \geq 0$ such that $\max_{0\leq k\leq \ell} \|\alpha_k^\ell\| \leq K$.

Next, by taking norms, squaring, and taking expectation on both sides of (10), we show that

$$\mathbb{E}\|\bar{\Delta}_t\|^2 \leq \frac{3K^2\|\Delta_0\|^2}{t^2} + \frac{3}{t^2}\mathbb{E}\left\|\sum_{k=0}^{t-2} \alpha_k^{t-1} v\rho_k\right\|^2 + \frac{3}{t^2}\mathbb{E}\left\|\sum_{k=0}^{t-2} \alpha_k^{t-1} M_{k+1}\right\|^2. \quad (11)$$

To bound the last term, we do the following sequence of steps. First, we use $\mathbb{E}[M_{k_2+1}|\mathcal{F}_{k_1}] = 0$ to show that

$$\mathbb{E}[(\alpha_{k_1}^{k-1} M_{k_1+1}^\top)^\top \alpha_{k_2}^{k-1} M_{k_2+1}] = 0$$

for $k_1 \neq k_2$. Next, we note that

$$\mathbb{E}\|M_{t+1}\|^2 \leq C_M[1 + \mathbb{E}\|\Delta_t\|^2 + \mathbb{E}\rho_t^2]$$

for a suitable constant $C_M \geq 0$. We then use these two relations to bound the last term in (11), which finally leads to the last term on the RHS in (9). Finally, we bound the second term on the RHS of (11) to get the third term on the RHS of (9). The core idea is to leverage Lemma 4.3 and an induction on $k_2 - k_1$ to get

$$|\mathbb{E}[\rho_{k_1}\rho_{k_2}]| \leq \frac{4R_{\max}^2}{Nk_2}$$

for $1 \leq k_1 \leq k_2$.

Proof of Theorem 3.1. It follows by combining the inequalities obtained in Lemmas 4.1, 4.3, 4.5, and 4.6. □

Remark 4.7: Although an optimal convergence rate of $O(\frac{1}{T})$ was obtained for $\mathbb{E}\rho_T^2$ and $\mathbb{E}\|\Delta_T\|^2$ in the *single-agent average-reward* setting even in [10], those results rely on stepsizes that depend on unknown model parameters. On the other hand, as shown in [29] which concerns the *single-agent discounted* case, it is possible to get rid of this restriction and obtain a convergence rate of $O(\frac{1}{T^\beta})$ using a universal stepsize $1/(t+1)^\beta$. However, this only works for $\beta < 1$ giving a sub-optimal convergence rate. We overcome this sub-optimality by additionally using Polyak-Ruppert averaging. Specifically, we show that the mean squared error of the averaged iterates in our setup, i.e., $\mathbb{E}\|\bar{\Delta}_T\|^2$, decays at the rate of $O(\frac{1}{NT})$. Note that the additional terms of $O(\frac{1}{(T+1)^2})$ in Theorem 3.1 decay fast enough to not have significant impact.

V. EXPERIMENTS

In this section, we present some illustrative numerical results. Each experiment below is specified by the number N of agents and the heterogeneity bounds ϵ_p and ϵ_r. The state and action spaces are common among the agents with $|\mathcal{S}| = |\mathcal{A}| = 100$. Further, the feature matrix $\Phi \in \mathbb{R}^{100\times 20}$, i.e., $d = 20$. The local reward and transition functions at all agents are chosen randomly while following the heterogeneity bounds (see Assumption $\mathcal{A}_2$). The number of agents considered are $N = 2, 5, 10$, and 20, while the heterogeneity bounds are either $\epsilon_p = \epsilon_r = 0.1$ or $\epsilon_p = \epsilon_r = 0.7$.

We compare two policy evaluation algorithms: our proposed method (Algorithm 1), and the single-timescale algorithm described in Section III, which is obtained by naively combining the ideas from [17] and [10]. We shall refer to the latter as NaiveFedTD(0). For Algorithm 1, we use stepsize $1/(t+1)$ for updating r_t and stepsize $1/(t+1)^\beta$ with $\beta \in (0, 1)$ for updating θ_t. For NaiveFedTD(0), as suggested in [10], we set $\beta_t := c_1/(c_2 + t)$ with $2 < \Delta c_1 < 2c_2$ and $c_\alpha > \Delta + \sqrt{1/\Delta^2 - 1}$, where $\Delta := \min_{\theta\in\mathbb{R}^d} \theta^\top A\theta$. Recall from Section III that, in NaiveFedTD(0), the r_t^i estimates are updated using the stepsize $c_\alpha\beta_t$, while θ_t is updated using β_t. We emphasize again that this c_1, c_2, and c_α choices are impractical since A depends on the different $\mathcal{P}_i^\mu$'s which are a priori unknown. However, if this knowledge is assumed, then the speedup, in terms of sample complexity, that this algorithm achieves is linear and the convergence rate optimal. In that sense, it serves as a benchmark for our algorithm.

Our first set of experiments consists of comparing the mean squared error as a function of the number of iterations for the two aforementioned algorithms. Fig. 1 gives the plots for various choices of N, ϵ_r, and ϵ_p. As can be seen, the two algorithms achieve similar decay rate and a speedup that is linear in the number of agents. In other words, the proposed Algorithm 1 achieves similar performance as NaiveFedTD(0) without relying on unknown problem parameters.

We note that the stepsize in Algorithm 1 depends on β. In the plots in Fig. 1, this parameter was not optimized. Instead, the parameter was chosen through a trial and error method to mirror the performance of NaiveFedTD(0). Indeed, how do we optimally choose β is an open question. Fig. 2 show the evolution of mean squared error for two heterogeneous setups when this parameter varies. It is evident that β should be chosen to be a small value. However, this conjecture is yet to be examined analytically.

VI. CONCLUSIONS

We considered the FRL problem when the models at various nodes are different. It has been shown recently that even in this heterogeneous case, a linear speedup in sample complexity is achievable. However, such results are known to hold only when discounted reward functions are used, and stepsizes are chosen carefully in a manner that depends on the unknown system parameters. In this work, we considered the average reward case with linear function approximation. By proposing and analyzing a novel two-timescale variant of federated TD(0) learning, along with a

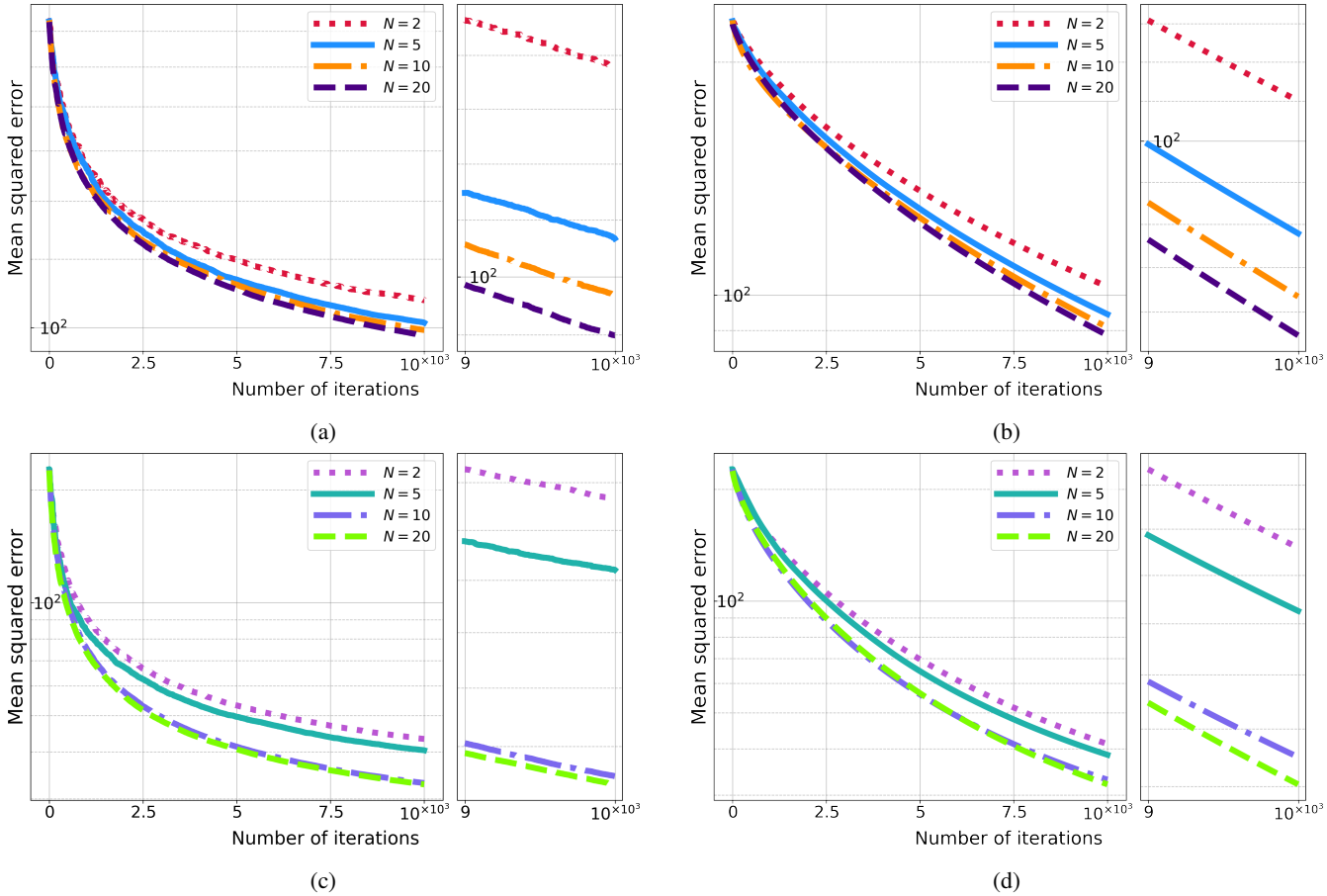

Fig. 1: Comparison of the performance of our proposed Algorithm 1 with that of NaiveFedTD(0) obtained by a naive combination of the methods in [17] and [10] as described in Section III. The y-axis shows the mean squared error, obtained by averaging over ten runs, while the x-axis shows the iteration number. For our proposed Algorithm 1, the mean squared error at iteration t refers to $\|\bar{\theta}_t - \theta_1^*\|^2$, and for NaiveFedTD(0) , it refers to $\|\theta_t - \theta_1^*\|^2$. The plots on the left depict the error decay of Algorithm 1, while the ones on the right depict the error decay of NaiveFedTD(0) . The plots on the top consider the choice $\epsilon_p = \epsilon_r = 0.1$ while the plots on the bottom consider the choice $\epsilon_p = \epsilon_r = 0.7$. The last 1000 iterations are magnified in the right of each plot. In both cases, the error decays faster with increasing N. Further, the performance of the two algorithms is comparable even though the naive single timescale algorithm requires knowledge of a priori unknown problem parameters. In this plot, Algorithm 1 has been run with an arbitrary choice of $\beta = 0.3$. Although the desired convergence rate of $O(\frac{1}{NT})$ is achieved, the choice of β is not optimized.

Polyak-Ruppert type averaging, we show that a linear speedup in sample complexity continues to hold even with a universal stepsize. For future work, we plan to extend these techniques to federated SARSA and federated Q-learning and explore tail-iterate averaging for potentially better convergence rates. Another aspect to study is the communication efficiency under iterate averaging.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[2] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, B. McMahan, *et al.*, "Towards federated learning at scale: System design," *Proceedings of machine learning and systems*, vol. 1, pp. 374–388, 2019.

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

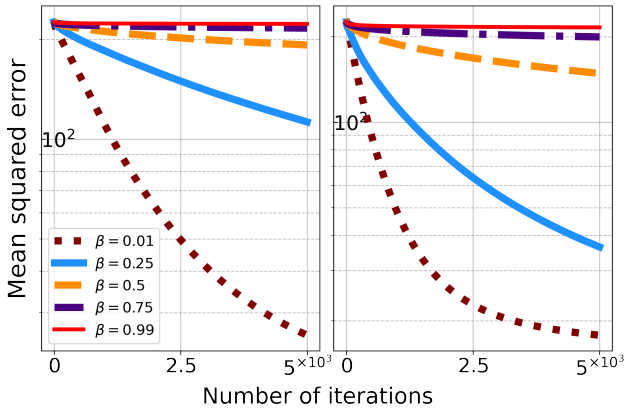

Fig. 2: Performance of our proposed Algorithm 1 for different choices of β. We consider the same setup as in Fig. 1 with $N = 5$. Each curve is the mean squared error pertaining to $\bar{\theta}_t$, obtained by averaging across 10 runs. The left subplot shows the mean squared error when heterogeneity parameters are $\epsilon_p = \epsilon_r = 0.1$ and the right one shows them when $\epsilon_p = \epsilon_r = 0.7$. In both cases, we observe that the performance improves with a decrease in the β value.

[4] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020.

[5] B. Recht, "A tour of reinforcement learning: The view from continuous control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 253–279, 2019.

[6] M. Cheng, C. Yin, J. Zhang, S. Nazarian, J. Deshmukh, and P. Bogdan, "A general trust framework for multi-agent systems," in *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 332–340, 2021.

[7] P. Kyriakis, J. V. Deshmukh, and P. Bogdan, "Specification mining and robust design under uncertainty: A stochastic temporal logic approach," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 18, no. 5s, pp. 1–21, 2019.

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

[9] A. Naik, R. Shariff, N. Yasui, and R. S. Sutton, "Discounted reinforcement learning is not an optimization problem," *CoRR*, vol. abs/1910.02140, 2019.

[10] S. Zhang, Z. Zhang, and S. T. Maguluri, "Finite sample analysis of average-reward td learning and q-learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 1230–1242, 2021.

[11] J. Qi, Q. Zhou, L. Lei, and K. Zheng, "Federated reinforcement learning: Techniques, applications, and open challenges," *ArXiv*, vol. abs/2108.11887, 2021.

[12] C. Nadiger, A. Kumar, and S. Abdelhak, "Federated reinforcement learning for fast personalization," in *2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pp. 123–127, IEEE, 2019.

[13] H. H. Zhuo, W. Feng, Y. Lin, Q. Xu, and Q. Yang, "Federated deep reinforcement learning," *arXiv preprint arXiv:1901.08277*, 2019.

[14] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-iid data with reinforcement learning," in *IEEE INFOCOM 2020-IEEE conference on computer communications*, pp. 1698–1707, IEEE, 2020.

[15] N. Dal Fabbro, A. Mitra, and G. J. Pappas, "Federated td learning over finite-rate erasure channels: Linear speedup under markovian sampling," *IEEE Control Systems Letters*, vol. 7, pp. 2461–2466, 2023.

[16] H. Jin, Y. Peng, W. Yang, S. Wang, and Z. Zhang, "Federated reinforcement learning with environment heterogeneity," 2022.

[17] H. Wang, A. Mitra, H. Hassani, G. J. Pappas, and J. Anderson, "Federated TD learning with linear function approximation under environmental heterogeneity," *Transactions on Machine Learning Research*, 2024.

[18] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM journal on control and optimization*, vol. 30, no. 4, pp. 838–855, 1992.

[19] D. Ruppert, *Stochastic approximation*. In Handbook of Sequetial Analysis, 1991.

[20] G. Patil, L. Prashanth, D. Nagaraj, and D. Precup, "Finite time analysis of temporal difference learning with linear function approximation: Tail averaging and regularisation," in *International Conference on Artificial Intelligence and Statistics*, pp. 5438–5448, PMLR, 2023.

[21] V. S. Borkar, "Stochastic approximation with two time scales," *Systems & Control Letters*, vol. 29, no. 5, pp. 291–294, 1997.

[22] A. Mokkadem and M. Pelletier, "Convergence rate and averaging of nonlinear two-time-scale stochastic approximation algorithms," *Annals of Applied Probability*, vol. 16, no. 3, pp. 1671–1702, 2006.

[23] G. Dalal, G. Thoppe, B. Szörényi, and S. Mannor, "Finite sample analysis of two-timescale stochastic approximation with applications to reinforcement learning," in *Conference On Learning Theory*, pp. 1199–1233, PMLR, 2018.

[24] G. Dalal, B. Szorenyi, and G. Thoppe, "A tale of two-timescale reinforcement learning with the tightest finite-time bound," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 3701–3708, 2020.

[25] M. Kaledin, E. Moulines, A. Naumov, V. Tadic, and H.-T. Wai, "Finite time analysis of linear two-timescale stochastic approximation with markovian noise," in *Conference on Learning Theory*, pp. 2144–2203, PMLR, 2020.

[26] S. Ganesh, A. Reiffers-Masson, and G. Thoppe, "Online learning with adversaries: A differential-inclusion analysis," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 1288–1293, IEEE, 2023.

[27] R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora, "Fast gradient-descent methods for temporal-difference learning with linear function approximation," in *Proceedings of the 26th annual international conference on machine learning*, pp. 993–1000, 2009.

[28] V. S. Borkar, *Stochastic approximation: a dynamical systems viewpoint*, vol. 48. Springer, 2009.

[29] G. Dalal, B. Szorenyi, G. Thoppe, and S. Mannor, "Finite sample analyses for td(0) with function approximation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 04 2018.