Collaborative Value Function Estimation Under Model Mismatch: A Federated Temporal Difference Analysis

Ali Beikmohammadi[1] (✉)[0000−0003−4884−4600], Sarit Khirirat[2][0000−0003−4473−2011], Peter Richtárik[2][0000−0003−4380−5848], and Sindri Magnússon[1][0000−0002−6617−8683]

[1] Department of Computer and Systems Sciences, Stockholm University, SE-164 25 Stockholm, Sweden
{beikmohammadi, sindri.magnusson}@dsv.su.se
[2] King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia
{sarit.khirirat, peter.richtarik}@kaust.edu.sa

Abstract. Federated reinforcement learning (FedRL) enables collaborative learning while preserving data privacy by preventing direct data exchange between agents. However, many existing FedRL algorithms assume that all agents operate in *identical environments*, which is often unrealistic. In real-world applications, such as multi-robot teams, crowd-sourced systems, and large-scale sensor networks, each agent may experience slightly different transition dynamics, leading to inherent model mismatches. In this paper, we first establish *linear convergence* guarantees for single-agent temporal difference learning ($\mathsf{TD(0)}$) in *policy evaluation* and demonstrate that under a perturbed environment, the agent suffers a systematic bias that prevents accurate estimation of the true value function. This result holds under both *i.i.d.* and *Markovian* sampling regimes. We then extend our analysis to the federated TD(0) ($\mathsf{FedTD(0)}$) setting, where multiple agents, each interacting with its own perturbed environment, *periodically* share value estimates to collaboratively approximate the true value function of a common underlying model. Our theoretical results indicate the impact of model mismatch, network connectivity, and mixing behavior on the convergence of $\mathsf{FedTD(0)}$. Empirical experiments corroborate our theoretical gains, highlighting that even moderate levels of information sharing significantly mitigate environment-specific errors.

Keywords: Federated Reinforcement Learning · Model Mismatch in Reinforcement Learning · Temporal Difference Learning · Policy Evaluation.

1 Introduction

Reinforcement learning (RL) has been widely applied in various domains, including robotics, healthcare, finance, and game playing, where agents must learn

to make sequential decisions in uncertain environments [43]. An agent in RL interacts with an unknown environment, observing states, taking actions, and receiving rewards, with the objective of learning an optimal policy that maximizes cumulative rewards. A key challenge in RL is learning the value function efficiently, especially when interactions with the environment are costly or time-consuming [4–6, 45].

Recently, *federated reinforcement learning* (FedRL) has emerged as a promising framework to improve sample efficiency and reduce learning time by allowing multiple agents to learn in parallel while exchanging information [37,58]. In a typical FedRL setup, multiple agents operate independently in separate instances of an environment, collect data locally, and communicate periodically their learned value estimates or policies to a central server or peer-to-peer network [25, 50]. By aggregating these estimates, FedRL can improve learning efficiency and robustness, particularly in distributed applications that exhibit data privacy or communication constraints [22, 52].

However, existing literature in FedRL usually assumes that all agents interact with *identical* environments [2, 17, 25, 29, 39]. In practice, this assumption often fails to hold. In multi-robot teams, different robots may have slightly different actuators, sensors, or physical constraints, thus leading to variations in transition dynamics [56]. In crowdsourced RL tasks, different users experience distinct environments due to network conditions, device capabilities, or regional differences. Even in large-scale sensor networks, environmental fluctuations can introduce discrepancies in the transition dynamics observed by different sensors. These variations lead to *model mismatch*, where each agent experiences a perturbed version of the *true* environment [31, 36].

In this paper, we study whether agents operating under *perturbed* transition dynamics can still collaboratively learn the true value function. Specifically, we focus on *policy evaluation*, which plays a fundamental role in RL as a precursor to improve the policy [7,42,46]. Our central research question is: *Can agents collaboratively learn the true value function while each leveraging information from a potentially different noisy model?* The presence of model mismatch introduces systematic bias in the learned value function, which does not necessarily vanish with more iterations [24,35,49,54]. While policy evaluation is a natural starting point, recent works [20, 40] have shown that with minor modifications, similar analysis techniques can extend to control settings.

Contributions. Our work provides a comprehensive theoretical analysis of temporal difference (TD) learning [42,43] under model mismatch in both *single-agent* and *federated* settings, under both *i.i.d.* and *Markovian* sampling regimes:

- First, we analyze **single-agent TD(0) under model mismatch**. Under both *i.i.d.* and *Markovian* sampling regimes, single-agent TD(0) attains linear convergence with the model mismatch error, which does not vanish as the algorithm progresses or when the step size decreases.
- We extend our analysis to **federated TD(0) (FedTD(0))**, where N agents, each with a different transition kernel, periodically exchange value estimates.

FedTD(0) achieves linear convergence with the model mismatch error that is reduced by the aggregation of value functions evaluated by multiple agents. Furthermore, our convergence bounds explicitly depend on the number of agents, degree of heterogeneity, and communication frequency.

– Finally, we corroborate our results with **numerical experiments**, showing that even moderate levels of collaboration significantly reduce bias and accelerate convergence to the true value function. Our experiments' code is publicly available at `https://github.com/AliBeikmohammadi/FedRL`.

2   Related Work

2.1   Single-agent TD Learning Algorithms

Much of the existing literature on TD learning has focused on the single-agent setting. Foundational studies proved the *asymptotic* convergence of on-policy TD methods with function approximation, leveraging stochastic approximation theory [10, 44, 47]. More recent advances have generalized these guarantees to off-policy learning scenarios [30,55]. In parallel, a separate line of research has established *finite-sample* or *non-asymptotic* guarantees for TD learning. In the on-policy setting, TD learning was studied under i.i.d. sampling regime by [16,27], and Markovian sampling regime by [9,12,23,32,38,40]. In the off-policy setting, non-asymptotic convergence of TD learning was also derived in [11,12]. Notably, most of these studies assume linear function approximation to leverage techniques from stochastic approximation. Our work departs from these single-agent analyses by examining a *multi-agent federated* framework, where each agent interacts with its own perturbed environment and periodically shares updates with others. This setting poses additional challenges related to agent heterogeneity, multiple dynamics, and intermittent communication, thereby requiring a specialized analytical approach.

2.2   Distributed and Federated RL Algorithms

Distributed RL. Many distributed RL algorithms have emerged to address scalability and efficiency challenges. This progress is exemplified by the development of many distributed frameworks, such as asynchronous parallelization [34], the high-throughput IMPALA architecture [19], and decentralized gossip-based protocols [1]. Theoretical convergence properties of such distributed methods have expanded to include robustness against adversarial attacks [22, 52] and communication compression [33]. Furthermore, several works derived sample complexities under i.i.d. sampling for distributed RL with linear function approximation [17, 29], and actor-critic algorithms [14, 39]. Further work extends to decentralized stochastic approximation [41,48,57], TD learning with linear function approximation [49], and off-policy TD actor-critic algorithms [13]. However, these analyses commonly rely on two key assumptions: agents synchronize updates through continuous communication (e.g., after every local iteration) and operate in *homogeneous* environments with identical dynamical properties across all participants.

FedRL under Homogeneous Environments. Parallel efforts in FedRL aim to reduce communication costs by allowing agents to perform multiple local updates between periodic synchronization rounds. This paradigm has been explored in contexts such as federated TD learning with linear function approximation [15, 21, 25], off-policy TD methods [25], and Q-learning [25], with further extensions to behavior-policy heterogeneity [51] and compressed communication [2]. However, a common limitation across all these approaches remains persistent: all agents are assumed to operate in identical Markov decision processes (MDPs) (*homogeneous* environments). This assumption fails to capture real-world scenarios where agents face environmental heterogeneity.

FedRL under Heterogeneous Environments. To our knowledge, only four works have studied FedRL under *heterogeneous* environments—a more closely related to our setting [24, 49, 53, 54], but each differs substantially from ours. For instance, [24] focuses on tabular Q-learning, and [53] studies the asymptotic behavior of distributed Q-learning. Meanwhile, [54] considers SARSA with linear function approximation, modeling data heterogeneity via a worst-case total-variation distance between transition kernels. The most closely related work is [49], which analyzes TD learning under Markovian sampling and periodic communication in a heterogeneous setting, but the analysis relies on function approximation, employs a more restrictive multiplicative bound on transition-model deviation, and uses a norm-induced approach that differs from ours. We depart from these approaches by considering a tabular federated policy evaluation problem in which each agent's environment is drawn from a distribution centered on the true transition model, and the agents periodically communicate with a central server. Our framework handles both i.i.d. and Markovian sampling, and it explicitly quantifies how inter-agent collaboration mitigates the bias introduced by environment perturbations in estimating the true value function. To our knowledge, this is the first systematic treatment of *model mismatch* in a tabular RL setting for both the single-agent and federated scenarios.

3 Model and Background

Within this section, we formalize the MDP setting and key definitions.

Discounted Infinite-horizon MDP. We study a discounted infinite-horizon MDP formally defined as the tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$. Here, $\mathcal{S}$ and $\mathcal{A}$ are finite state and action spaces, respectively; $\mathcal{P}$ represents a set of the action-dependent transition probabilities; $\mathcal{R}$ is a bounded reward function; $\gamma \in (0, 1)$ is the discount factor governing long-term reward trade-offs.

Value Function and Bellman Operator. Under a fixed policy $\mu \colon \mathcal{S} \to \mathcal{A}$, the MDP reduces to a Markov reward process (MRP) with transition matrix P_μ and reward function $r := R_\mu$. Here, $P_\mu(s, s')$ is the probability of transitioning from s to s' under action $\mu(s)$, and $r(s)$ denotes the expected immediate reward

at state s. Our goal is to evaluate the value function V, which captures the expected discounted return when starting from any state s and following μ:

$$V(s) \;=\; \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t\, r(s^{(t)}) \mid s^{(0)} = s \Big], \quad \forall s \in \mathcal{S}, \tag{1}$$

where the expectation is taken over trajectories generated by the transition kernel P_μ (i.e., $s^{(t+1)} \sim P_\mu\left(\cdot \mid s^{(t)}\right)$) for all $t \geq 0$.

A fundamental result in dynamic programming [7, 8] states that V is the unique fixed point of the policy-specific Bellman operator $T \colon \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$, defined as:

$$\big(TV\big)(s) \;=\; r(s) \;+\; \gamma \sum_{s' \in \mathcal{S}} P_\mu(s, s')\, V(s') \quad \forall s \in \mathcal{S}. \tag{2}$$

Equivalently, V satisfies the Bellman equation: $TV = V$.

4 Single-Agent TD(0) under Model Mismatch

One popular approach for estimating the value function V in a model-free setting (i.e., when the underlying MDP is unknown) is the TD(0) algorithm [42, 43]. At each time step t, the agent observes a state $s^{(t)}$, receives reward $r^{(t)} := r(s^{(t)})$, transitions to a next state $s^{(t+1)}$, and updates its current value estimate $V^{(t)}(\cdot)$ by modifying only the coordinate at $s^{(t)}$. Specifically, TD(0) updates via:

$$V^{(t+1)}\big(s^{(t)}\big) \;=\; V^{(t)}\big(s^{(t)}\big) + \alpha\,\delta^{(t)} \;\; \text{and} \;\; \delta^{(t)} = r^{(t)} + \gamma\,V^{(t)}\big(s^{(t+1)}\big) - V^{(t)}\big(s^{(t)}\big), \tag{3}$$

where $\alpha \in (0,1)$ is a step size, and $V^{(0)}$ is some initial guess. For states $s \neq s^{(t)}$, we simply set $V^{(t+1)}(s) = V^{(t)}(s)$. Under standard assumptions—such as a sufficiently small step size α, bounded rewards, and ergodic state sampling—TD(0) converges to the unique fixed point of the Bellman operator T, which is precisely the true value function V of the policy μ [10, 43, 47].

However, this holds when the agent interacts with the true transition probability P_μ, which we aim to evaluate the policy for. Here, we instead consider scenarios where the agent interacts with a perturbed environment $\hat{P}$ rather than P_μ, as explained in Algorithm 1. Specifically, we analyze the convergence of the single-agent TD(0) algorithm under the following model mismatch assumption.

Assumption 1. *Let the empirical transition matrix $\hat{P}$ be a perturbed transition matrix of P_μ, which satisfies for $\Delta > 0$*

$$\left\| \hat{P} - P_\mu \right\|_2^2 \leq \Delta^2. \tag{4}$$

Assumption 1 implies the deviation of the empirical transition matrix $\hat{P}$ from the true transition matrix P_μ. Small values of Δ imply that $\hat{P}$ is close to P_μ. Furthermore, $\hat{P}$ can possibly be a biased estimate of P_μ.

Next, we analyze the convergence of the single-agent TD(0) algorithm under model mismatch with both well-known sampling regimes, i.i.d. and Markovian, in Sections 4.1 and 4.2, respectively.

Algorithm 1 Single-agent TD(0)

Initialize: Learning rate $\alpha \in (0,1)$, Discount factor $\gamma \in (0,1)$, Number of communication rounds T, Initial value function $V^{(0)}$.

for each time step $t = 0, 1, \ldots, T-1$ **do**
    Observe state $s^{(t)}$ according to the chosen sampling regime (i.i.d. or Markovian).
    Receive $r^{(t)}$ and get $s^{(t+1)} \sim \hat{P}(\cdot | s^{(t)})$.
    Update $V^{(t+1)}$ via

$$V^{(t+1)}(s^{(t)}) = V^{(t)}(s^{(t)}) + \alpha[r^{(t)} + \gamma V^{(t)}(s^{(t+1)}) - V^{(t)}(s^{(t)})].$$

end for

4.1   Analysis for I.I.D. Setting

In the i.i.d. sampling regime, each state $s^{(t)}$ is independently drawn from the stationary distribution over $\mathcal{S}$. Although $s^{(t)}$ and $s^{(t-1)}$ are uncorrelated, we still generate a next state $s^{(t+1)} \sim \hat{P}(\cdot \mid s^{(t)})$ in TD(0) update (3) to perform one-step bootstrapping. Hence, while the *tuples* $(s^{(t)}, s^{(t+1)})$ are sampled i.i.d. from the stationary distribution, the actual *visited* states do not form a Markov chain. The i.i.d. sampling regime eliminates temporal correlations in the sampled states, and therefore simplifies the convergence analysis.

The following theorem establishes the linear convergence of single-agent TD(0) under i.i.d. sampling.

Theorem 1 (Single-agent, i.i.d. sampling). *Consider the single-agent TD(0) algorithm (Algorithm 1) under the i.i.d. sampling. Let a row-stochastic transition matrix $\hat{P}$ satisfy Assumption 1. Then, with probability at least $1 - \delta$,*

$$\|e^{(t)}\|_2 \leq [1 - \alpha(1-\gamma)]^t \|e^{(0)}\|_2 + \frac{\gamma \Delta \sqrt{|\mathcal{S}|}}{(1-\gamma)^2} + \frac{\alpha \sqrt{t}}{(1-\gamma)} \sqrt{32(\log(t/\delta) + 1/4)},$$

where $e^{(t)} := V^{(t)} - V$, $|\mathcal{S}|$ is the number of states, and $\alpha \in (0,1)$.

Remark 1. From Theorem 1, the single-agent TD(0) algorithm ensures linear convergence with a high probability of its value function $V^{(t)}$ towards the unique true value function V with the first error term due to the model mismatch Δ and the second error term of $\frac{\alpha \sqrt{t}}{(1-\gamma)} \sqrt{32(\log(t/\delta) + 1/4)}$ due to the i.i.d. sampling regime. Decreasing the step size α cannot reduce the first error term due to the model mismatch, while decreasing the second error term due to i.i.d. sampling at the price of a slow convergence rate. For instance, if we choose $\alpha = 1/T^\eta$ with $\eta \in (1/2, 1)$, $0 < t \leq T$, where T is the given number of total iteration counts, then the algorithm converges at the rate:

$$\|e^{(t)}\|_2 \leq \exp\left(-\frac{t(1-\gamma)}{T^\eta}\right) \|e^{(0)}\|_2 + \mathcal{O}\left(\frac{\sqrt{t}\log(t)}{T^\eta}\right) + \frac{\gamma \Delta \sqrt{|\mathcal{S}|}}{(1-\gamma)^2},$$

with probability at least $1 - \delta$. In conclusion, the convergence bound for the algorithm contains the model mismatch error term $\frac{\gamma \Delta \sqrt{|\mathcal{S}|}}{(1-\gamma)^2}$.

4.2   Analysis for Markovian Setting

As a more realistic setting, we also study Algorithm 1 under a Markovian sampling regime. In this regime, the states follow a Markov chain defined by the transition matrix $\hat{P}$. Concretely, $s^{(t+1)} \sim \hat{P}(\cdot \mid s^{(t)})$ for each t, matching our earlier definition of MRP induced by a fixed policy. Although this better captures real-world dynamics, analyzing the corresponding TD(0) updates becomes more intricate due to the temporal correlations in $\{s^{(t)}\}$. To analyze the single-agent TD(0) algorithm under the Markovian sampling regime, we make an assumption on the Markov chain and define the mixing time τ_ϵ in the single-agent setting.

Assumption 2. *The Markov chain induced by the policy μ is aperiodic and irreducible.*

Definition 1. *Let τ_ϵ be the minimum time such that the following holds:* $\|\xi^{(t)}\|_2 \leq \epsilon, \forall t \geq \tau_\epsilon$. *Here, $\xi^{(t)} \in \mathbb{R}^{|\mathcal{S}|}$ is defined by $\xi^{(t)}(s) = \delta^{(t)} - \mathbb{E}\big[\delta^{(t)} \mid \mathcal{F}^{(t)}\big]$ when $s = s^{(t)}$, and $\xi^{(t)}(s) = 0$ for all other s, where $\mathcal{F}^{(t)}$ is the filtration up to t.*

This assumption implies that the Markov chain induced by μ admits a unique stationary distribution π, and mixes at a geometric rate [28]. The consequence of this assumption is that there exists some $T \geq 1$ such that $\tau_\epsilon \leq T \log(1/\epsilon)$.

The next theorem shows the linear convergence of the single-agent TD(0) algorithm under Markovian sampling.

Theorem 2 (Single-agent, Markovian sampling). *Consider the single-agent TD(0) algorithm (Algorithm 1) under Markovian sampling. Let a row-stochastic transition matrix $\hat{P}$ satisfy Assumption 1 and the Markov chain satisfy Assumption 2. Then,*

$$\|e^{(t)}\|_2 \leq [1 - \alpha(1 - \gamma)]^t \|e^{(0)}\|_2 + \gamma \frac{\Delta\sqrt{|\mathcal{S}|}}{(1 - \gamma)^2} + \frac{\alpha}{1 - \gamma}\left(2\tau_\alpha + 1\right),$$

where $e^{(t)} := V^{(t)} - V$, $|\mathcal{S}|$ is the number of states, $\alpha \in (0, 1)$, and τ_α is defined by Definition 1.

Remark 2. The single-agent TD(0) algorithm under the Markovian sampling regime, similar to the i.i.d. sampling regime, achieves the linear convergence of its value function $V^{(t)}$ towards the fixed value function V with two residual error terms. In particular, decreasing the step size α cannot reduce the first error term due to the model mismatch Δ, while decreasing the second error term due to the mixing time τ_α of the aperiodic and irreducible Markov chain at the price of slow convergence. For instance, the algorithm with $\alpha = 1/T^\eta$ for $\eta \in (1/2, 1)$ converges at the rate

$$\|e^{(t)}\|_2 \leq \exp\left(-\frac{t(1 - \gamma)}{T^\eta}\right)\|e^{(0)}\|_2 + \mathcal{O}\left(\frac{1}{T^\eta}\right) + \frac{\gamma\Delta\sqrt{|\mathcal{S}|}}{(1 - \gamma)^2}.$$

In conclusion, the single-agent TD(0) algorithm under both i.i.d sampling and Markovian sampling regimes converges with the model mismatch error term

$\frac{\gamma \Delta \sqrt{|\mathcal{S}|}}{(1-\gamma)^2}$ that cannot be reduced by decreasing the step size. In the next section, we show that this model mismatch error term can be reduced by the benefits of multiple agents for running the TD(0) algorithm in the federated setting.

5   Extension to FedTD(0) under Model Mismatch

To show the benefits of multiple agents, we extend our results to the federated setting. We consider the FedTD(0) algorithm, where multiple agents collaboratively estimate the value function. In each communication round $t = 0, 1, \ldots, T-1$, each agent $i = 1, 2, \ldots, N$ receives the global value function estimate $V^{(t)}$ from the server, sets $V_i^{(t,0)} = V^{(t)}$, and updates its local estimate $V_i^{(t,k)}$ according to:

$$V_i^{(t,k+1)}(s_i^{(t,k)}) = V_i^{(t,k)}(s_i^{(t,k)}) + \alpha \delta_i^{(t,k)}, \quad \text{and}$$
$$\delta_i^{(t,k)} = r_i^{(t,k)} + \gamma V_i^{(t,k)}(s_i^{(t,k+1)}) - V_i^{(t,k)}(s_i^{(t,k)}) \text{ for } k = 0, 1, \ldots, K-1.$$

Here, the step size is denoted by $\alpha \in (0, 1)$, $s_i^{(t,k)}$ is the observed state, $r_i^{(t,k)}$ is the received reward for each agent, and $s_i^{(t,k+1)}$ is drawn from the agent's transition probability conditioned on $s_i^{(t,k)}$. Then, the central server computes the average of the received estimate progress from all the agents $\frac{1}{N} \sum_{i=1}^{N}(V_i^{(t,K)} - V^{(t)})$, and updates the global value function estimate via:

$$V^{(t+1)} = V^{(t)} + \frac{\beta}{N} \sum_{i=1}^{N}(V_i^{(t,K)} - V^{(t)}),$$

where $\beta \in (0, 1]$ is the federated tuning parameter. The description of FedTD(0) algorithm is provided in Algorithm 2. To demonstrate the benefit of multiple agents in FedTD(0), we impose the following model mismatch assumption.

Assumption 3. *Let the empirical transition matrices $\hat{P}_1, \hat{P}_2, \ldots, \hat{P}_N$ be the perturbed matrix of P_μ, which satisfies Assumption 1 with $\Delta > 0$, and also*

$$\left\| \frac{1}{N} \sum_{i=1}^{N} \hat{P}_i - P_\mu \right\|_2^2 \leq \frac{\Lambda^2}{N}. \tag{5}$$

This assumption implies that as the number of agents N grows, the average model $(1/N) \sum_{i=1}^{N} \hat{P}_i$ converges to the true model P_μ at the rate $\mathcal{O}(1/N)$. This captures the intuition that if agents have similar dynamics on average, their collective behavior also becomes more predictable. In particular, this assumption captures well when $\hat{P}_i$ are row-stochastic, i.e. $\|\hat{P}_i\|_2 \leq \|\hat{P}_i\|_1 \leq 1$, and are sampled under both i.i.d. and Markovian sampling regimes. On the one hand, if $\hat{P}_i$ are sampled under i.i.d. sampling, and satisfy $\mathbb{E}\|\hat{P}_i\|_2 = P_\mu$ and $\|\hat{P}_i\|_2 \leq 1$, then according to Lemma A.2. of [18], we obtain (5) with $\Lambda = [6(1+\sqrt{\log(1/\delta)})]^2$ with

---

Algorithm 2 FedTD(0)

Initialize: Learning rate $\alpha \in (0,1)$, Federated parameter $\beta \in (0,1]$, Discount factor $\gamma \in (0,1)$, Number of agents N, Number of local steps K, Number of communication rounds T, Initial value function $V^{(0)}$.

for each communication round $t = 0, 1, \ldots, T-1$ **do**

    for each agent $i = 1, 2, \ldots, N$ **in parallel do**

        Set $V_i^{(t,0)} = V^{(t)}$, where $V^{(t)}$ is the global value estimate from the server.

        for $k = 0, 1, \ldots, K-1$ **do**

            Observe state $s_i^{(t,k)}$ according to the chosen sampling (i.i.d. or Markovian).

            Receive $r_i^{(t,k)}$ and get $s_i^{(t,k+1)} \sim \hat{P}_i(\cdot|s_i^{(t,k)})$.

            Update $V_i^{(t,k+1)}$ via

$$V_i^{(t,k+1)}(s_i^{(t,k)}) = V_i^{(t,k)}(s_i^{(t,k)}) + \alpha[r_i^{(t,k)} + \gamma V_i^{(t,k)}(s_i^{(t,k+1)}) - V_i^{(t,k)}(s_i^{(t,k)})].$$

        end for

        Send $V_i^{(t,k+1)} - V^{(t)}$ back to the server.

    end for

    Server computes and broadcasts global

$$V^{(t+1)} = V^{(t)} + \frac{\beta}{N}\sum_{i=1}^{N}(V_i^{(t,K)} - V^{(t)}).$$

end for

---

probability at least $1 - \delta$. On the other hand, if $\hat{P}_i$ are sampled under Markovian sampling and satisfy $\|\hat{P}_i\|_2 \leq 1$, and the Markov chain satisfies Assumption 2 with the mixing time τ_ϵ, then according to Lemma A.6 of [18] we have (5) with $\Lambda = \tilde{\mathcal{O}}\left(\tau_\epsilon \lceil 2\log(N)\rceil\right)$ in expectation.

5.1 Analysis for I.I.D. Setting

FedTD(0) under i.i.d. sampling achieves linear convergence in high probability with a lower residual error than single-agent TD(0), as shown next.

Theorem 3 (Federated, i.i.d. sampling). *Consider FedTD(0) (Algorithm 2) under i.i.d. sampling. Let each row-stochastic transition matrix $\hat{P}_i$ of the ith-agent satisfy Assumption 3. Then, with probability at least $1 - \delta$,*

$$\|e^{(t)}\|_2 \leq \rho^t \|e^{(0)}\|_2 + \frac{B_1}{\sqrt{N}} + \alpha^2 B_2 + \frac{4}{\sqrt{N}}\frac{\beta\alpha\sqrt{t}\sqrt{K}[A(\delta/(3Kt))]^3}{(1-\gamma)},$$

where $e^{(t)} := V^{(t)} - V$, $\rho = (1-\beta) + \beta[(1-\alpha) + \alpha\gamma]^K$, $B_1 = \gamma\frac{\Lambda\sqrt{|\mathcal{S}|}}{(1-\gamma)^2(1-[1-\alpha(1-\gamma)]^K)}$, $B_2 = \gamma^2\frac{C\Delta\sqrt{|\mathcal{S}|}}{(1-\gamma)(1-[1-\alpha(1-\gamma)]^K)}$, $A(\delta) = \sqrt{2(\log(1/\delta + 1/4)}$, $C = \exp(K(C_P + C_\mu))$ where $\|\hat{P}_i^l\|_2 \leq C_P$ and $\|P_\mu^l\|_2 \leq C_\mu$ for $l, C_P, C_\mu \geq 0$, and $\beta, \alpha \in (0,1)$.

Remark 3. FedTD(0) under i.i.d. sampling achieves high-probability convergence at the linear rate with the $\frac{B_1}{\sqrt{N}}$ error term due to the model mismatch Λ, the $\alpha^2 B_2$ error term due to the model mismatch Δ, and the $\frac{4}{\sqrt{N}} \frac{\beta\alpha\sqrt{t}\sqrt{K}[A(\delta/(3Kt))]^3}{(1-\gamma)}$ error term due to i.i.d. sampling. Unlike the single-agent case from Theorem 1, FedTD(0) achieves the $\sqrt{N}$-speedup, where N is the number of agents.

Remark 4. The step size α, the federated parameter β, the number of local steps K, and the number of agents N impact the convergence rate and residual error terms of FedTD(0) under i.i.d. sampling. We can reduce the error term due to i.i.d. sampling at the price of slow convergence, either by decreasing α and β. For instance, we can reduce the error term due to i.i.d. sampling by choosing $\beta = \frac{1}{T^\eta}$ and $\alpha = \frac{1}{K^\eta}$ with $\eta \in (1/2, 1)$, which yields

$$\frac{\beta\alpha\sqrt{t}\sqrt{K}[A(\delta/(3t))]^3}{(1-\gamma)} = \mathcal{O}\left(\frac{\sqrt{t}(\log(Kt/\delta))^{3/2}}{T^\eta}\frac{1}{K^{\eta-1/2}}\right).$$

Furthermore, the error term due to Δ and Λ can be decreased only by reducing α and only by increasing the number of agents N, respectively.

Remark 5. Under i.i.d. sampling, FedTD(0) can be shown to achieve a significantly more accurate value function $V^{(t)}$ than single-agent TD(0). We can show this by proving that FedTD(0) attains lower residual errors than single-agent TD(0). First, two error terms due to Λ and i.i.d. sampling of FedTD(0), unlike single-agent TD(0), vanish, as N approaches $+\infty$. Second, the error term due to Δ of FedTD(0) can be proved to be lower than single-agent TD(0) by a factor of $1/(1-\gamma)$ by setting $K \to +\infty$ and $\alpha = \sqrt{\frac{1}{\gamma C}}$ in Theorem 3. This yields

$$\alpha^2\frac{\gamma^2 C\Delta\sqrt{|\mathcal{S}|}}{(1-\gamma)(1-[1-\alpha(1-\gamma)]^K)} \approx \alpha^2\gamma^2\frac{C\Delta\sqrt{|\mathcal{S}|}}{(1-\gamma)} = \frac{\gamma\Delta\sqrt{|\mathcal{S}|}}{(1-\gamma)} \overset{\gamma\in(0,1)}{\leq} \frac{\gamma\Delta\sqrt{|\mathcal{S}|}}{(1-\gamma)^2}.$$

In conclusion, as N grows, FedTD(0) under i.i.d. sampling from Theorem 3 achieves the lower residual error than single-agent TD(0) from Theorem 1.

5.2   Analysis for Markovian Setting

Finally, we establish the convergence of FedTD(0) under Markovian sampling. To show this, we define the mixing time in the federated setting as follows.

Definition 2. *Let τ_ϵ be the minimum time such that the following holds: For any agent, $k \geq 0$ and $t \geq \tau_\epsilon$, $\|\xi_i^{(t,k)}\|_2 \leq \epsilon$. Here, $\xi_i^{(t,k)} \in \mathbb{R}^{|\mathcal{S}|}$ is defined by $\xi_i^{(t,k)}(s) = \delta_i^{(t,k)} - \mathbb{E}\big[\delta_i^{(t,k)} \mid \mathcal{F}^{(t,k)}\big]$ when $s = s_i^{(t,k)}$, and $\xi_i^{(t,k)} = 0$ for all other s, where $\mathcal{F}^{(t,k)}$ is the filtration up to iterations t, k.*

We establish the convergence theorem of FedTD(0) under Markovian sampling, which achieves the same $\sqrt{N}$-speedup as FedTD(0) under i.i.d. sampling.

Theorem 4 (Federated, Markovian sampling). *Consider* FedTD(0) *(Algorithm 2) under i.i.d. sampling. Let each row-stochastic transition matrix* $\hat{P}_i$ *of the* i^{th}-*agent satisfy Assumption 3. Then,*

$$\|e^{(t)}\|_2 \leq \rho^t \|e^{(0)}\|_2 + \frac{B_1}{\sqrt{N}} + \alpha^2 B_2 + \beta \frac{1}{1-\gamma} \left(\frac{2\tau_\beta}{1-\gamma} + t\beta \right),$$

where $e^{(t)} := V^{(t)} - V$, $\rho = (1-\beta) + \beta[(1-\alpha) + \alpha\gamma]^K$, $B_1 = \gamma \frac{\Lambda\sqrt{|S|}}{(1-\gamma)^2(1-[1-\alpha(1-\gamma)]^K)}$, $B_2 = \gamma^2 \frac{C\Delta\sqrt{|S|}}{(1-\gamma)(1-[1-\alpha(1-\gamma)]^K)}$, $C = \exp(K(C_P + C_\mu))$ *where* $\|\hat{P}_i^l\|_2 \leq C_P$ *and* $\|P_\mu^l\|_2 \leq C_\mu$ *for* $l, C_P, C_\mu \geq 0$, τ_β *is defined by Definition 2, and* $\beta, \alpha \in (0,1)$.

Remark 6. FedTD(0) under Markovian sampling, similar to i.i.d. sampling, attains linear convergence with the $\frac{B_1}{\sqrt{N}}$ error term due to the model mismatch Λ, the $\alpha^2 B_2$ error term due to the model mismatch Δ, and the $\beta\frac{1}{1-\gamma}\left(\frac{2\tau_\beta}{1-\gamma} + t\beta\right)$ error term due to Markovian sampling τ_β. Unlike FedTD(0) under i.i.d. sampling, which achieves the $\sqrt{N}$-speedup for two error terms due to Λ and i.i.d. sampling, FedTD(0) under Markovian sampling attains the $\sqrt{N}$-speedup only for the error term due to Λ.

Remark 7. The step size α, the federated parameter β, the number of local steps K, and the number of agents N influence the convergence rate and residual error terms of FedTD(0) under Markovian sampling. First, we can reduce the error term due to Δ at the price of slow convergence by decreasing α. Second, decreasing β lessens the error term due to Markovian sampling τ_β at the cost of slow convergence. For instance, the error term due to Markovian sampling can be lessened by choosing $\beta = \frac{1}{T^\eta}$ with $\eta \in (1/2, 1)$, as

$$\beta\frac{1}{(1-\gamma)}\left(\frac{2\tau_\beta}{1-\gamma} + t\beta\right) = \mathcal{O}\left(\frac{1}{T^\eta}\right) + \mathcal{O}\left(\frac{1}{T^{2\eta-1}}\right).$$

Third, increasing the number of agents N decreases the error term due to Λ.

Remark 8. Under Markovian sampling, like i.i.d. sampling, FedTD(0) can be shown to achieve higher accuracy of value functions $V^{(t)}$ than single-agent TD(0). We show this by proving that under certain conditions, two residual error terms of FedTD(0) due to Λ and Δ from Theorem 4 are lower than single-agent TD(0) from Theorem 2. First, the error term due to Λ for FedTD(0) vanishes, as the number of agents N goes to $+\infty$. Second, the error term due to Δ for FedTD(0) is lower than single-agent TD(0) by a factor of $1/(1-\gamma)$ for $\gamma \in (0,1)$ by choosing $K \to +\infty$ and $\alpha = \sqrt{\frac{1}{\gamma C}}$.

6 Experiments

6.1 Setup

We evaluate the empirical performance of FedTD(0) under varying levels of model mismatch. We consider a randomly generated MDP with 10 states, where
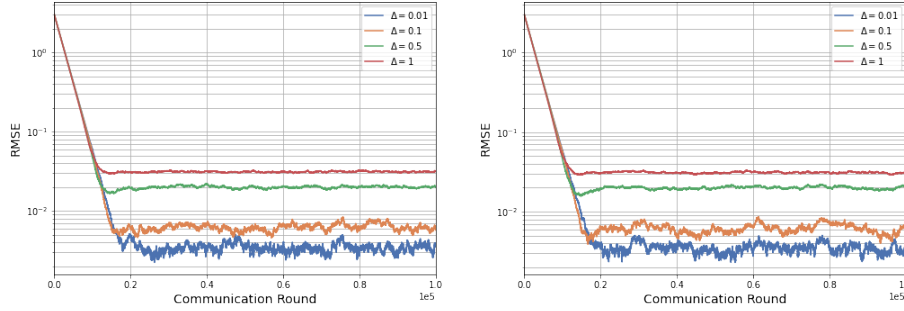
Fig. 1: Impact of the model mismatch Δ on the RMSE of the V-estimates for FedTD(0) with (**Left**) i.i.d sampling and (**Right**) Markovian sampling. Here, $N = 10$, $K = 5$, $\alpha = 0.01$, and $\beta = 0.4$.

the transition matrix P_μ is row-stochastic and generated using uniform distributions. The reward function r also has a uniform $[0, 1]$ distribution. To simulate heterogeneity among agents, we introduce small perturbations to the transition matrix P_μ, ensuring that each agent i interacts with a slightly different transition model $\hat{P}_i$. The perturbation is controlled by a parameter Δ, such that the Frobenius norm difference satisfies $\|\hat{P}_i - P_\mu\|_2 \leq \Delta$. This perturbation is followed by a projection step to ensure that each $\hat{P}_i$ remains row-stochastic. The reward function is kept identical across agents to isolate the effect of transition dynamics mismatch. To quantify the error in value function estimation, we compute the root mean square error (RMSE) between the global value estimate $V^{(t)}$ and the true value function V, where V is computed via the Bellman fixed-point equation $(I - \gamma P_\mu)V = r$. We track the evolution of this metric over communication rounds t. The value function is initialized with zero entries ($V^{(0)} = 0$ for all s), and we set $\gamma = 0.8$. All experiments are repeated with five different seed numbers, and results are averaged for robustness.

6.2   Discussion

Effect of Model Mismatch under I.I.D. and Markovian Sampling. The model mismatch Δ affects the accuracy of value function estimates under both i.i.d. and Markovian sampling. From Figure 1, increasing Δ leads to a higher RMSE in the estimated value function. Interestingly, the residual error behaves almost identically in both sampling regimes, as further illustrated in Appendix D.1, Figure 5 and Appendix D.2, Figure 6. This confirms our theoretical analysis in Theorems 3 and 4, which state that while the convergence dynamics differ under i.i.d. and Markovian sampling, the final bias due to model mismatch is primarily governed by Δ. In particular, a large value of Δ results in a large residual error around the unique fixed value function V. These results highlight that FedRL methods must account for model mismatch effects rather than focusing solely on the sampling strategy of individual agents.
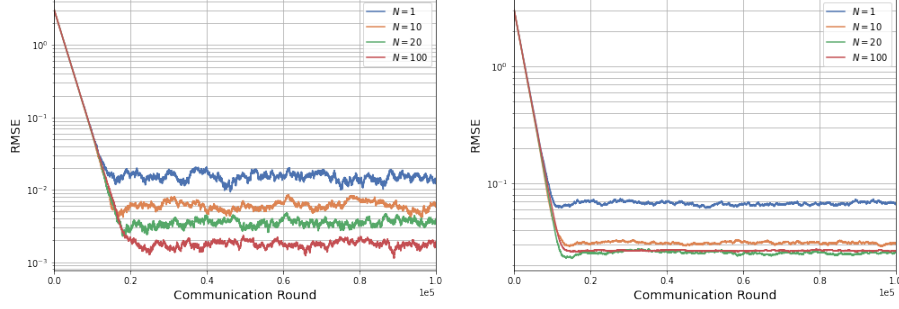
Fig. 2: Impact of the number of agents N on the RMSE of V-estimates for FedTD(0) with Markovian sampling. Here, $K = 5$, $\alpha = 0.01$, and $\beta = 0.4$. (**Left**) Corresponds to a model mismatch of $\Delta = 0.1$, while (**Right**) considers a more severe mismatch of $\Delta = 1$.
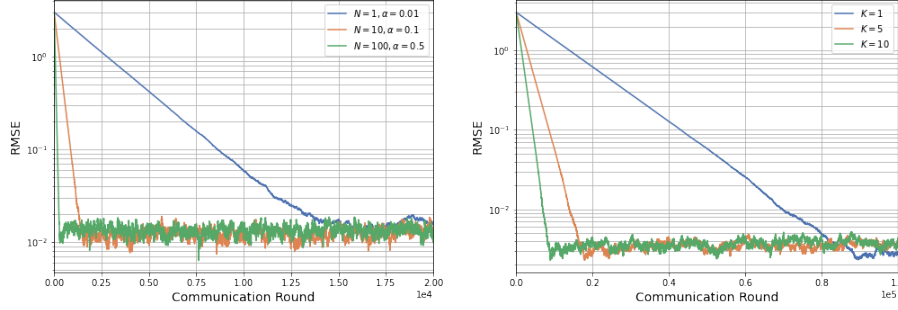

Fig. 3: RMSE of the V-estimates for FedTD(0) with Markovian sampling. (**Left**) Examines the effect of the number of agents N and the learning rate α with $K = 5$. (**Right**) Studies the effect of the number of local steps K while keeping $N = 20$ and $\alpha = 0.01$. Both cases use $\Delta = 0.1$ and $\beta = 0.4$.

Effect of the Number of Agents on Reducing Model Mismatch Bias. From Figure 2, FedTD(0) with the larger number of agents N ensures that its estimated value function is closer to the true value function. This corroborates the $\sqrt{N}$-speedup in the convergence in Theorems 3 and 4. Further validation of this trend is provided in Appendix D.3, Figure 7.

Convergence Speedup with More Agents. As shown in Figure 3, more agents allow for a larger step size α without destabilizing the learning process. This is because federated averaging reduces variance in updates, enabling more aggressive updates without divergence. Consequently, FedTD(0) with more agents achieves similar accuracy in fewer iterations compared to a single-agent setting, making it a scalable solution for distributed RL applications. This aligns with
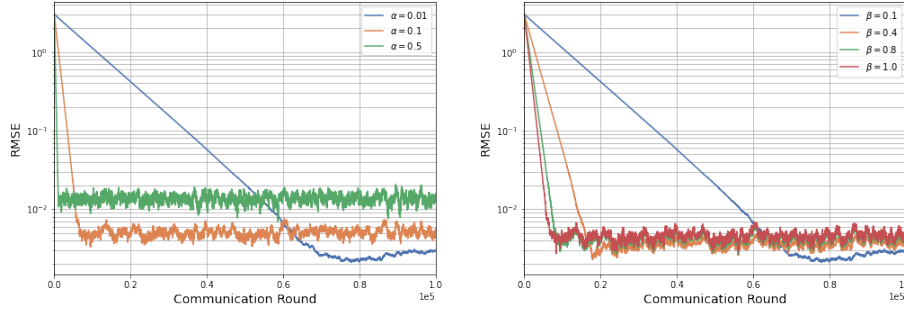
Fig. 4: RMSE of the V-estimates for FedTD(0) with Markovian sampling. (**Left**) Examines the effect of the learning rate α with $\beta = 0.1$. (**Right**) Studies the effect of the federated parameter β while $\alpha = 0.01$. Both cases use $N = 20$, $K = 5$, and $\Delta = 0.1$.

the $\sqrt{N}$-speedup and the lower residual error by multiple agents from Theorems 3 and 4.

Robustness to the Choice of Local Steps K: Communication Efficiency. As shown in Figure 3, increasing K reduces communication overhead without significantly compromising final performance. Specifically, $K = 10$ achieves the same RMSE as $K = 1$, but with a *10-fold reduction in communication rounds*. This suggests that increasing local updates can substantially improve efficiency in FedRL, making it particularly useful in scenarios where the communication cost is expensive.

The Effect of Learning Rate α and Federated Parameter β. Figure 4 illustrates how α and β impact the performance of FedTD(0). A smaller α slows convergence but stabilizes learning, whereas a larger α leads to faster convergence. Similarly, a large β allows faster adaptation of the global value estimate. However, changes in α have a greater effect on residual error compared to adjustments in β, as supported by our non-asymptotic results (see Theorem 4), where the convergence of FedTD(0) consists of the two residual error terms, $\mathcal{O}(\alpha^2) + \mathcal{O}(\beta)$.

7   Conclusion

In this paper, we investigated FedTD(0) for *policy evaluation* under *model mismatch*, where multiple agents interact with perturbed environments and *periodically* exchange value estimates. We established *linear convergence* guarantees for single-agent TD(0) under both *i.i.d.* and *Markovian* sampling regimes and demonstrated how environmental perturbations introduce systematic bias in individual learning. Extending these results to the federated setting, we quantified the role of model mismatch, network connectivity, and mixing behavior in the convergence of FedTD(0). Our theoretical results indicate that even under heterogeneous transition dynamics, moderate levels of information sharing among

agents can effectively mitigate environment-specific errors and improve convergence rates. Empirical results further support these findings, demonstrating that federated collaboration reduces individual bias and accelerates convergence to the true value function. These findings highlight the potential of FedRL for real-world applications where identical environment assumptions are impractical, such as multi-robot coordination and decentralized control in sensor networks. For future research, we aim to extend our framework beyond policy evaluation to control settings, such as federated Q-learning, and explore the effectiveness of FedRL in real-world tasks.

Acknowledgments

Supplementary Material

For comprehensive supplementary appendices and accompanying code, readers are directed to the full version of this paper, accessible via Springer Link and arXiv [3], along with the corresponding code repository hosted on GitHub: `https://github.com/AliBeikmohammadi/FedRL`.

References

1. Assran, M., Romoff, J., Ballas, N., Pineau, J., Rabbat, M.: Gossip-based actor-learner architectures for deep reinforcement learning. Advances in Neural Information Processing Systems **32** (2019)
2. Beikmohammadi, A., Khirirat, S., Magnússon, S.: Compressed federated reinforcement learning with a generative model. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 20–37. Springer (2024)
3. Beikmohammadi, A., Khirirat, S., Richtárik, P., Magnússon, S.: Collaborative value function estimation under model mismatch: A federated temporal difference analysis (2025), `https://arxiv.org/abs/2503.17454`
4. Beikmohammadi, A., Magnússon, S.: Comparing NARS and reinforcement learning: An analysis of ONA and Q-learning algorithms. In: International Conference on Artificial General Intelligence. pp. 21–31. Springer (2023)

5. Beikmohammadi, A., Magnússon, S.: Ta-explore: Teacher-assisted exploration for facilitating fast reinforcement learning. In: Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems. pp. 2412–2414 (2023)

6. Beikmohammadi, A., Magnússon, S.: Accelerating actor-critic-based algorithms via pseudo-labels derived from prior knowledge. Information Sciences **661**, 120182 (2024)

7. Bellman, R.: Dynamic programming, vol. 153. American Association for the Advancement of Science (1966)

8. Bertsekas, D., Tsitsiklis, J.N.: Neuro-dynamic programming. Athena Scientific (1996)

9. Bhandari, J., Russo, D., Singal, R.: A finite time analysis of temporal difference learning with linear function approximation. In: Conference on learning theory. pp. 1691–1692. PMLR (2018)

10. Borkar, V.S.: Stochastic approximation: a dynamical systems viewpoint, vol. 48. Springer (2009)

11. Chen, Z., Maguluri, S.T., Shakkottai, S., Shanmugam, K.: Finite-sample analysis of contractive stochastic approximation using smooth convex envelopes. Advances in Neural Information Processing Systems **33**, 8223–8234 (2020)

12. Chen, Z., Maguluri, S.T., Shakkottai, S., Shanmugam, K.: A lyapunov theory for finite-sample guarantees of asynchronous Q-learning and TD-learning variants. arXiv preprint arXiv:2102.01567 (2021)

13. Chen, Z., Zhou, Y., Chen, R.R.: Multi-agent off-policy TDC with near-optimal sample and communication complexities. Transactions on Machine Learning Research (2022)

14. Chen, Z., Zhou, Y., Chen, R.R., Zou, S.: Sample and communication-efficient decentralized actor-critic algorithms with finite-time analysis. In: International Conference on Machine Learning. pp. 3794–3834. PMLR (2022)

15. Dal Fabbro, N., Mitra, A., Pappas, G.J.: Federated td learning over finite-rate erasure channels: Linear speedup under markovian sampling. IEEE Control Systems Letters (2023)

16. Dalal, G., Szörényi, B., Thoppe, G., Mannor, S.: Finite sample analyses for td (0) with function approximation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)

17. Doan, T., Maguluri, S., Romberg, J.: Finite-time analysis of distributed td (0) with linear function approximation on multi-agent reinforcement learning. In: International Conference on Machine Learning. pp. 1626–1635. PMLR (2019)

18. Dorfman, R., Levy, K.Y.: Adapting to mixing time in stochastic optimization with markovian data. In: International Conference on Machine Learning. pp. 5429–5446. PMLR (2022)

19. Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., et al.: Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In: International conference on machine learning. pp. 1407–1416. PMLR (2018)

20. Even-Dar, E., Mansour, Y., Bartlett, P.: Learning rates for q-learning. Journal of machine learning Research **5**(1) (2003)

21. Fabbro, N.D., Mitra, A., Heath, R., Schenato, L., Pappas, G.J.: Over-the-Air federated TD learning. In: MLSys 2023 Workshop on Resource-Constrained Learning in Wireless Networks (2023)

22. Fan, X., Ma, Y., Dai, Z., Jing, W., Tan, C., Low, B.K.H.: Fault-tolerant federated reinforcement learning with theoretical guarantee. In: Ranzato, M., Beygelzimer,

A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems. vol. 34, pp. 1007–1021. Curran Associates, Inc. (2021)

23. Hu, B., Syed, U.: Characterizing the exact behaviors of temporal difference learning algorithms using markov jump linear system theory. Advances in neural information processing systems **32** (2019)

24. Jin, H., Peng, Y., Yang, W., Wang, S., Zhang, Z.: Federated reinforcement learning with environment heterogeneity. In: International Conference on Artificial Intelligence and Statistics. pp. 18–37. PMLR (2022)

25. Khodadadian, S., Sharma, P., Joshi, G., Maguluri, S.T.: Federated reinforcement learning: Linear speedup under markovian sampling. In: International Conference on Machine Learning. pp. 10997–11057. PMLR (2022)

26. Kohler, J.M., Lucchi, A.: Sub-sampled cubic regularization for non-convex optimization. In: International Conference on Machine Learning. pp. 1895–1904. PMLR (2017)

27. Lakshminarayanan, C., Szepesvari, C.: Linear stochastic approximation: How far does constant step-size and iterate averaging go? In: International Conference on Artificial Intelligence and Statistics. pp. 1347–1355. PMLR (2018)

28. Levin, D.A., Peres, Y.: Markov chains and mixing times, vol. 107. American Mathematical Soc. (2017)

29. Liu, R., Olshevsky, A.: Distributed td (0) with almost no communication. IEEE Control Systems Letters **7**, 2892–2897 (2023)

30. Maei, H.R.: Convergent actor-critic algorithms under off-policy training and function approximation. arXiv preprint arXiv:1802.07842 (2018)

31. Mankowitz, D.J., Levine, N., Jeong, R., Abdolmaleki, A., Springenberg, J.T., Shi, Y., Kay, J., Hester, T., Mann, T., Riedmiller, M.: Robust reinforcement learning for continuous control with model misspecification. In: International Conference on Learning Representations (2020)

32. Mitra, A.: A simple finite-time analysis of td learning with linear function approximation. IEEE Transactions on Automatic Control (2024)

33. Mitra, A., Pappas, G.J., Hassani, H.: Temporal difference learning with compressed updates: Error-feedback meets reinforcement learning. Transactions on Machine Learning Research (2024)

34. Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: International conference on machine learning. pp. 1928–1937. PMLR (2016)

35. Morimoto, J., Doya, K.: Robust reinforcement learning. Neural computation **17**(2), 335–359 (2005)

36. Pinto, L., Davidson, J., Sukthankar, R., Gupta, A.: Robust adversarial reinforcement learning. In: International conference on machine learning. pp. 2817–2826. PMLR (2017)

37. Qi, J., Zhou, Q., Lei, L., Zheng, K.: Federated reinforcement learning: techniques, applications, and open challenges. Intelligence & Robotics **1**(1) (2021)

38. Samsonov, S., Tiapkin, D., Naumov, A., Moulines, E.: Improved high-probability bounds for the temporal difference learning algorithm via exponential stability. In: The Thirty Seventh Annual Conference on Learning Theory. pp. 4511–4547. PMLR (2024)

39. Shen, H., Zhang, K., Hong, M., Chen, T.: Towards understanding asynchronous advantage actor-critic: Convergence and linear speedup. IEEE Transactions on Signal Processing (2023)

40. Srikant, R., Ying, L.: Finite-time error bounds for linear stochastic approximation andtd learning. In: Conference on Learning Theory. pp. 2803–2830. PMLR (2019)

41. Sun, J., Wang, G., Giannakis, G.B., Yang, Q., Yang, Z.: Finite-time analysis of decentralized temporal-difference learning with linear function approximation. In: International Conference on Artificial Intelligence and Statistics. pp. 4485–4495. PMLR (2020)
42. Sutton, R.S.: Learning to predict by the methods of temporal differences. Machine learning **3**, 9–44 (1988)
43. Sutton, R.S., Barto, A.G., et al.: Reinforcement learning: An introduction, vol. 1. MIT press Cambridge (1998)
44. Tadić, V.: On the convergence of temporal-difference learning with linear function approximation. Machine learning **42**, 241–267 (2001)
45. Tesauro, G., et al.: Temporal difference learning and td-gammon. Communications of the ACM **38**(3), 58–68 (1995)
46. Tsitsiklis, J., Van Roy, B.: An analysis of temporal-difference learning with function approximation. IEEE Transactions on Automatic Control **42**(5), 674–690 (1997)
47. Tsitsiklis, J., Van Roy, B.: Analysis of temporal-diffference learning with function approximation. Advances in neural information processing systems **9** (1996)
48. Wai, H.T.: On the convergence of consensus algorithms with markovian noise and gradient bias. In: 2020 59th IEEE Conference on Decision and Control (CDC). pp. 4897–4902. IEEE (2020)
49. Wang, H., Mitra, A., Hassani, H., Pappas, G.J., Anderson, J.: Federated TD learning with linear function approximation under environmental heterogeneity. Transactions on Machine Learning Research (2024)
50. Woo, J., Joshi, G., Chi, Y.: The blessing of heterogeneity in federated Q-learning: Linear speedup and beyond. In: Proceedings of the 40th International Conference on Machine Learning. ICML'23, JMLR.org (2023)
51. Woo, J., Joshi, G., Chi, Y.: The blessing of heterogeneity in federated Q-learning: Linear speedup and beyond. In: International Conference on Machine Learning. pp. 37157–37216. PMLR (2023)
52. Wu, Z., Shen, H., Chen, T., Ling, Q.: Byzantine-resilient decentralized policy evaluation with linear function approximation. IEEE Transactions on Signal Processing **69**, 3839–3853 (2021)
53. Xie, Z., Song, S.: FedKL: Tackling data heterogeneity in federated reinforcement learning by penalizing KL divergence. IEEE Journal on Selected Areas in Communications **41**(4), 1227–1242 (2023)
54. Zhang, C., Wang, H., Mitra, A., Anderson, J.: Finite-time analysis of on-policy heterogeneous federated reinforcement learning. In: The Twelfth International Conference on Learning Representations (2024)
55. Zhang, S., Liu, B., Yao, H., Whiteson, S.: Provably convergent two-timescale off-policy actor-critic with function approximation. In: International Conference on Machine Learning. pp. 11204–11213. PMLR (2020)
56. Zhao, W., Queralta, J.P., Westerlund, T.: Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In: 2020 IEEE symposium series on computational intelligence (SSCI). pp. 737–744. IEEE (2020)
57. Zheng, Z., Gao, F., Xue, L., Yang, J.: Federated Q-learning: Linear regret speedup with low communication cost. In: The Twelfth International Conference on Learning Representations (2024)
58. Zhuo, H.H., Feng, W., Lin, Y., Xu, Q., Yang, Q.: Federated deep reinforcement learning. arXiv preprint arXiv:1901.08277 (2019)