Online Robust Reinforcement Learning with Model Uncertainty

Yue Wang
University at Buffalo
Buffalo, NY 14228
ywang294@buffalo.edu

Shaofeng Zou
University at Buffalo
Buffalo, NY 14228
szou3@buffalo.edu

Abstract

Robust reinforcement learning (RL) is to find a policy that optimizes the worst-case performance over an uncertainty set of MDPs. In this paper, we focus on *model-free* robust RL, where the uncertainty set is defined to be centering at a misspecified MDP that generates a single sample trajectory sequentially, and is assumed to be *unknown*. We develop a sample-based approach to estimate the unknown uncertainty set, and design robust Q-learning algorithm (tabular case) and robust TDC algorithm (function approximation setting), which can be implemented in an online and incremental fashion. For the robust Q-learning algorithm, we prove that it converges to the optimal robust Q function, and for the robust TDC algorithm, we prove that it converges asymptotically to some stationary points. Unlike the results in [Roy et al., 2017], our algorithms do not need any additional conditions on the discount factor to guarantee the convergence. We further characterize the finite-time error bounds of the two algorithms, and show that both the robust Q-learning and robust TDC algorithms converge as fast as their vanilla counterparts (within a constant factor). Our numerical experiments further demonstrate the robustness of our algorithms. Our approach can be readily extended to robustify many other algorithms, e.g., TD, SARSA, and other GTD algorithms.

1 Introduction

Existing studies on Markov decision process (MDP) and reinforcement learning (RL) [Sutton and Barto, 2018] mostly rely on the crucial assumption that the environment on which a learned policy will be deployed is the same one that was used to generate the policy, which is often violated in practice – e.g., the simulator may be different from the true environment, and the MDP may evolve over time. Due to such model deviation, the actual performance of the learned policy can significantly degrade. To address this problem, the framework of robust MDP was formulated in [Bagnell et al., 2001, Nilim and El Ghaoui, 2004, Iyengar, 2005], where the transition kernel of the MDP is not fixed and lies in an uncertainty set, and the goal is to learn a policy that performs well under the worst-case MDP in the uncertainty set. In [Bagnell et al., 2001, Nilim and El Ghaoui, 2004, Iyengar, 2005], it was assumed that the uncertainty set is known beforehand, i.e., model-based approach, and dynamic programming can be used to find the optimal robust policy.

The model-based approach, however, requires a model of the uncertainty set known beforehand, and needs a large memory to store the model when the state and action spaces are large, which make it less applicable for many practical scenarios. This motivates the study in this paper, *model-free* robust RL with model uncertainty, which is to learn a robust policy using a single sample trajectory from a misspecified MDP, e.g., a simulator and a similar environment in which samples are easier to collect than in the target environment where the policy is going to be deployed. The major challenge lies in that the transition kernel of the misspecified MDP is not given beforehand, and thus, the uncertainty

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

set and the optimal robust policy need to be learned simultaneously using sequentially observed data from the misspecified MDP. Moreover, robust RL learns the value function of the worst-case MDP in the uncertainty set which is different from the misspecified MDP that generates samples. This is similar to the off-policy learning, which we refer to as the "off-transition-kernel" setting. Therefore, the learning may be unstable and could diverge especially when function approximation is used [Baird, 1995].

In this paper, we develop a model-free approach for robust RL with model uncertainty. Our major contributions in this paper are summarized as follows.

- Motivated by empirical studies of adversarial training in RL [Huang et al., 2017, Kos and Song, 2017, Lin et al., 2017, Pattanaik et al., 2018, Mandlekar et al., 2017] and the R-contamination model in robust detection (called ϵ-contamination model in [Huber, 1965]), we design the uncertainty set using the R-contamination model (see (4) for the details). We then develop an approach to estimate the unknown uncertainty set using only the current sample, which does not incur any additional memory cost. Unlike the approach in [Roy et al., 2017], where the uncertainty set is relaxed to one not depending on the misspecified MDP that generates samples so that an online algorithm can be constructed, our approach does not need to relax the uncertainty set.
- We develop a robust Q-learning algorithm for the tabular case, which can be implemented in an online and incremental fashion, and has the same memory cost as the vanilla Q-learning algorithm. We show that our robust Q-learning algorithm converges asymptotically, and further characterize its finite-time error bound. Unlike the results in [Roy et al., 2017] where a stringent condition on the discount factor (which is due to the relaxation of the uncertainty set, and prevents the use of a discount factor close to 1 in practice) is needed to guarantee the convergence, our algorithm converges without the need of such condition. Furthermore, our robust Q-learning algorithm converges as fast as the vanilla Q-learning algorithm [Li et al., 2020] (within a constant factor), while being robust to model uncertainty.
- We generalize our approach to the case with function approximation (for large state/action space). We investigate the robust policy evaluation problem, i.e., evaluate a given policy under the worst-case MDP in the uncertainty set. As mentioned before, the robust RL problem is essentially "off-transition-kernel", and therefore non-robust methods with function approximation may diverge [Baird, 1995] (also see our experiments). We develop a novel extension of the gradient TD (GTD) method [Maei et al., 2010, Maei, 2011, Sutton et al., 2008] to robust RL. Our approach introduces a novel smoothed robust Bellman operator to construct the smoothed mean-squared projected robust Bellman error (MSPRBE). Using our uncertainty set design and online sample-based estimation, we develop a two time-scale robust TDC algorithm. We further characterize its convergence and finite-time error bound.
- We conduct numerical experiments to validate the robustness of our approach. In our experiments, our robust Q-learning algorithm achieves a much higher reward than the vanilla Q-learning algorithm when being trained on a misspecified MDP; and our robust TDC algorithm converges much faster than the vanilla TDC algorithm, and the vanilla TDC algorithm may even diverge.

1.1 Related Work

Model-Based Robust MDP. The framework of robust MDP was investigated in [Iyengar, 2005, Nilim and El Ghaoui, 2004, Bagnell et al., 2001, Satia and Lave Jr, 1973, Wiesemann et al., 2013], where the transition kernel is assumed to be in some uncertainty set, and the problem can be solved by dynamic programming. This approach was further extended to the case with function approximation in [Tamar et al., 2014]. However, these studies are model-based, which assume beforehand knowledge of the uncertainty set. In this paper, we investigate the model-free setting, where the uncertainty set is a set of MDPs centered around some unknown Markov transition kernel from which a single sample trajectory can be sequentially observed.

Adversarial Robust RL. It was shown in [Iyengar, 2005] that the robust MDP problem is equivalent to a zero-sum game between the agent and the nature. Motivated by this fact, the adversarial training approach, where an adversary perturbs the state transition, was studied in [Vinitsky et al., 2020, Pinto et al., 2017, Abdullah et al., 2019, Hou et al., 2020, Rajeswaran et al., 2017, Atkeson and Morimoto, 2003, Morimoto and Doya, 2005]. This method relies on a simulator, where the state transition can be modified in an arbitrary way. Another approach is to modify the current state through adversarial

samples, which is more heuristic, e.g., [Huang et al., 2017, Kos and Song, 2017, Lin et al., 2017, Pattanaik et al., 2018, Mandlekar et al., 2017]. Despite the empirical success of these approaches, theoretical performance guarantees, e.g., convergence to the optimal robust policy and convergence rate, are yet to be established. The main difference lies in that during the training, our approach does not need to manipulate the state transition of the MDP. More importantly, we develop the asymptotic convergence to the optimal robust policy and further characterize the finite-time error bound. In [Lim et al., 2013], the scenario where some unknown parts of the state space can have arbitrary transitions while other parts are purely stochastic was studied. Adaptive algorithm to adversarial behavior was designed, and its regret bound is shown to be similar to the purely stochastic case. In [Zhang et al., 2020a], the robust adversarial RL problem for the special linear quadratic case was investigated.

Model-free Robust RL. In [Roy et al., 2017, Badrinath and Kalathil, 2021] model-free RL with model uncertainty was studied, where in order to construct an algorithm that can be implemented in an online and incremental fashion, the uncertainty set was firstly relaxed by dropping the dependency on the misspecified MDP that generates the samples (centroid of the uncertainty set). Such a relaxation is pessimistic since the relaxed uncertainty set is not centered at the misspecified MDP anymore (which is usually similar to the target MDP), making the robustness to the relaxed uncertainty set not well-justified. Such a relaxation will further incur a stringent condition on the discounted factor to guarantee the convergence, which prevents the use of a discount factor close to 1 in practice. Moreover, only asymptotic convergence was established in [Roy et al., 2017]. In this paper, we do not relax the uncertainty set, and instead propose an online approach to estimate it. Our algorithms converge without the need of the condition on the discount factor. We also provides finite-time error bounds for our algorithms. The multi-agent RL robust to reward uncertainty was investigated in [Zhang et al., 2020b], where the reward uncertainty set is known, but the transition kernel is fixed.

Finite-time Error Bound for RL Algorithms. For the tubular case, Q-learning has been studied intensively, e.g., in [Even-Dar et al., 2003, Beck and Srikant, 2012, Qu and Wierman, 2020, Li et al., 2020, Wainwright, 2019, Li et al., 2021]. TD with function approximation were studied in [author=Dalal, Gal and Szörényi, Balázs and Thoppe, Gugan and Mannor, Shie, 2018, Bhandari et al., 2018, Srikant and Ying, 2019, Cai et al., 2019, Sun et al., 2020]. Q-learning and SARSA with linear function approximation were investigated in [Zou et al., 2019, Chen et al., 2019]. The finite-time error bounds for the gradient TD algorithms [Maei et al., 2010, Sutton et al., 2009, Maei et al., 2010] were further developed recently in [Dalal et al., 2018, Liu et al., 2015, Gupta et al., 2019, Xu et al., 2019, Dalal et al., 2020, Kaledin et al., 2020, Ma et al., 2020, Wang and Zou, 2020, Ma et al., 2021, Doan, 2021]. There are also finite-time error bounds on the policy gradient methods and actor critic methods, e.g., [Wang et al., 2020, Yang et al., 2019, Kumar et al., 2019, Qiu et al., 2019, Wu et al., 2020, Cen et al., 2020, Bhandari and Russo, 2019, Agarwal et al., 2021, Mei et al., 2020]. We note that these studies are for the non-robust RL algorithms, and in this paper, we design robust RL algorithms, and characterize their finite-time error bounds.

2 Preliminaries

Markov Decision Process. An MDP can be characterized by a tuple $(\mathcal{S}, \mathcal{A}, \mathsf{P}, c, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, $\mathsf{P} = \left\{p_s^a \in \Delta_{|\mathcal{S}|}, a \in \mathcal{A}, s \in \mathcal{S}\right\}$ is the transition kernel[1], c is the cost function, and $\gamma \in [0, 1)$ is the discount factor. Specifically, p_s^a denotes the distribution of the next state if taking action a at state s. Let $p_s^a = \{p_{s,s'}^a\}_{s' \in \mathcal{S}}$, where $p_{s,s'}^a$ denotes the probability that the environment transits to state s' if taking action a at state s. The cost of taking action a at state s is given by $c(s, a)$. A stationary policy π is a mapping from $\mathcal{S}$ to a distribution over $\mathcal{A}$. At each time t, an agent takes an action $A_t \in \mathcal{A}$ at state $S_t \in \mathcal{S}$. The environment then transits to the next state S_{t+1} with probability $p_{S_t,S_{t+1}}^{A_t}$, and the agent receives cost given by $c(S_t, A_t)$. The value function of a policy π starting from any initial state $s \in \mathcal{S}$ is defined as the expected accumulated discounted cost by following π: $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t c(S_t, A_t) | S_0 = s, \pi\right]$, and the goal is to find the policy π that minimizes the above value function for any initial state $s \in \mathcal{S}$.

Robust Markov Decision Process. In the robust case, the transition kernel is not fixed and lies in some uncertainty set. Denote the transition kernel at time t by P_t, and let $\kappa = (\mathsf{P}_0, \mathsf{P}_1, ...)$, where $\mathsf{P}_t \in \mathbf{P}, \forall t \geq 0$, and $\mathbf{P}$ is the uncertainty set of the transition kernel. The sequence κ can be viewed as the policy of the nature, and is adversarially chosen by the nature [Bagnell et al., 2001, Nilim and

[1] Δ_n denotes the $(n-1)$-dimensional probability simplex: $\{(p_1, ..., p_n) | 0 \leq p_i \leq 1, \sum_{i=1}^n p_i = 1\}$.

El Ghaoui, 2004, Iyengar, 2005]. Define the robust value function of a policy π as the worst-case expected accumulated discounted cost following a fixed policy π over all transition kernels in the uncertainty set:

$$V^{\pi}(s) = \max_{\kappa} \mathbb{E}_{\kappa} \left[\sum_{t=0}^{\infty} \gamma^t c(S_t, A_t) | S_0 = s, \pi \right], \tag{1}$$

where $\mathbb{E}_{\kappa}$ denotes the expectation when the state transits according to κ. Similarly, define the robust action-value function for a policy π: $Q^{\pi}(s,a) = \max_{\kappa} \mathbb{E}_{\kappa} \left[\sum_{t=0}^{\infty} \gamma^t c(S_t, A_t) | S_0 = s, A_0 = a, \pi \right]$. The goal of robust RL is to find the optimal robust policy π^* that minimizes the worst-case accumulated discounted cost:

$$\pi^* = \arg\min_{\pi} V^{\pi}(s), \forall s \in \mathcal{S}. \tag{2}$$

We also denote V^{π^*} and Q^{π^*} by V^* and Q^*, respectively, and $V^*(s) = \min_{a \in \mathcal{A}} Q^*(s,a)$.

Note that a transition kernel is a collection of conditional distributions. Therefore, the uncertainty set $\mathbf{P}$ of the transition kernel can be equivalently written as a collection of $\mathcal{P}_s^a$ for all $s \in \mathcal{S}, a \in \mathcal{A}$, where $\mathcal{P}_s^a$ is a set of conditional distributions p_s^a over the state space $\mathcal{S}$. Denote by $\sigma_{\mathcal{P}}(v) \triangleq \max_{p \in \mathcal{P}} (p^{\top} v)$ the support function of vector v over a set of probability distributions $\mathcal{P}$. For robust MDP, the following robust analogue of the Bellman recursion was provided in [Nilim and El Ghaoui, 2004, Iyengar, 2005].

Theorem 1. *[Nilim and El Ghaoui, 2004] The following perfect duality condition holds for all $s \in \mathcal{S}$:*

$$\min_{\pi} \max_{\kappa} \mathbb{E}_{\kappa} \left[\sum_{t=0}^{\infty} \gamma^t c(S_t, A_t) \big| \pi, S_0 = s \right] = \max_{\kappa} \min_{\pi} \mathbb{E}_{\kappa} \left[\sum_{t=0}^{\infty} \gamma^t c(S_t, A_t) \big| \pi, S_0 = s \right]. \tag{3}$$

The optimal robust value function V^* satisfies $V^*(s) = \min_{a \in \mathcal{A}} (c(s,a) + \gamma \sigma_{\mathcal{P}_s^a}(V^*))$, and the optimal robust action-value function Q^* satisfies $Q^*(s,a) = c(s,a) + \gamma \sigma_{\mathcal{P}_s^a}(V^*)$.

Define the robust Bellman operator $\mathbf{T}$ by $\mathbf{T}Q(s,a) = c(s,a) + \gamma \sigma_{\mathcal{P}_s^a}(\min_{a \in \mathcal{A}} Q(s,a))$. It was shown in [Nilim and El Ghaoui, 2004, Iyengar, 2005] that $\mathbf{T}$ is a contraction and its fixed point is the optimal robust Q^*. When the uncertainty set is known, so that $\sigma_{\mathcal{P}_s^a}$ can be computed exactly, V^* and Q^* can be solved by dynamic programming [Iyengar, 2005, Nilim and El Ghaoui, 2004].

3 R-Contamination Model For Uncertainty Set Construction

In this section, we construct the uncertainty set using the R-contamination model.

Let $\mathsf{P} = \{p_s^a, s \in \mathcal{S}, a \in \mathcal{A}\}$ be the centroid of the uncertainty set, i.e., the transition kernel that generates the sample trajectory, and P is *unknown*. For example, P can be the simulator at hand, which may not be exactly accurate; and P can be the transition kernel of environment 1, from which we can take samples to learn a policy that will be deployed in a similar environment 2. The goal is to learn a policy using samples from P that performs well when applied to a perturbed MDP from P.

Motivated by empirical studies of adversarial training in RL [Huang et al., 2017, Kos and Song, 2017, Lin et al., 2017, Pattanaik et al., 2018, Mandlekar et al., 2017] and the R-contamination model in robust detection [Huber, 1965], we use the R-contamination model to define the uncertainty set:

$$\mathcal{P}_s^a = \left\{ (1-R)p_s^a + Rq | q \in \Delta_{|\mathcal{S}|} \right\}, s \in \mathcal{S}, a \in \mathcal{A}, \text{ for some } 0 \le R \le 1. \tag{4}$$

Here, p_s^a is the centroid of the uncertainty set $\mathcal{P}_s^a$ at (s,a), which is *unknown*, and R is the design parameter of the uncertainty set, which measures the size of the uncertainty set, and is assumed to be known in the algorithm. We then let $\mathbf{P} = \bigotimes_{s \in \mathcal{S}, a \in \mathcal{A}} \mathcal{P}_s^a$.

Remark 1. *R-contamination model is closely related to other uncertainty set models like total variation and KL-divergence. It can be shown that R-contamination set certered at p is a subset of total variation ball : $\left\{ (1-R)p + Rq | q \in \Delta_{|\mathcal{S}|} \right\} \subset \left\{ q \in \Delta_{|\mathcal{S}|} | d_{TV}(p,q) \le R \right\}$. Hence the total variation uncertainty set is less conservative than our R-contamination uncertainty set. KL-divergence moreover can be related to total variation using Pinsker's inequality, i.e., $d_{TV}(p,q) \le \sqrt{\frac{1}{2} d_{KL}(p,q)}$.*

4 Tabular Case: Robust Q-Learning

In this section, we focus on the tabular case with finite state and action spaces. We focus on the asynchronous setting where a single sample trajectory is available with Markovian noise. We will develop an efficient approach to estimate the unknown uncertainty set $\mathbf{P}$, and further the support function $\sigma_{\mathcal{P}_s^a}(\cdot)$, and then design our robust Q-learning algorithm.

We propose an efficient and data-driven approach to estimate the unknown p_s^a and thus the unknown uncertainty set $\mathcal{P}_s^a$ for any $s \in \mathcal{S}$ and $a \in \mathcal{A}$. Specifically, denote the sample at t-th time step by $O_t = (s_t, a_t, s_{t+1})$. We then use O_t to obtain the maximum likelihood estimate (MLE) $\hat{p}_t \triangleq \mathbb{1}_{s_{t+1}}$ of the transition kernel $p_{s_t}^{a_t}$, where $\mathbb{1}_{s_{t+1}}$ is a probability distribution taking probability 1 at s_{t+1} and 0 at other states. This is an unbiased estimate of the transition kernel $p_{s_t}^{a_t}$ conditioning on $S_t = s_t$ and $A_t = a_t$. We then design a sample-based estimate $\hat{\mathcal{P}}_t \triangleq \left\{(1-R)\hat{p}_t + Rq | q \in \Delta_{|\mathcal{S}|}\right\}$ of the uncertainty set $\mathcal{P}_{s_t}^{a_t}$. Using the sample-based uncertainty set $\hat{\mathcal{P}}_t$, we construct the following robust Q-learning algorithm in Algorithm 1. For any t, $\sigma_{\hat{\mathcal{P}}_t}(V_t)$ can be easily computed: $\sigma_{\hat{\mathcal{P}}_t}(V_t) =$

Algorithm 1 Robust Q-Learning

Initialization: T, $Q_0(s,a)$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, behavior policy π_b, s_0, step size α_t

1: **for** $t = 0, 1, 2, ..., T-1$ **do**
2: Choose a_t according to $\pi_b(\cdot|s_t)$
3: Observe s_{t+1} and c_t
4: $V_t(s) \leftarrow \min_{a\in\mathcal{A}} Q_t(s,a), \forall s \in \mathcal{S}$
5: $Q_{t+1}(s_t,a_t) \leftarrow (1-\alpha_t)Q_t(s_t,a_t) + \alpha_t(c_t + \gamma\sigma_{\hat{\mathcal{P}}_t}(V_t))$
6: $Q_{t+1}(s,a) \leftarrow Q_t(s,a)$ for $(s,a) \neq (s_t,a_t)$
7: **end for**

Output: Q_T

$R\max_{s\in\mathcal{S}} V_t(s) + (1-R)V_t(s_{t+1})$. Hence the update in Algorithm 1 (line 5) can be written as

$$Q_{t+1}(s_t,a_t) \leftarrow (1-\alpha_t)Q_t(s_t,a_t) + \alpha_t(c_t + \gamma R \max_{s\in\mathcal{S}} V_t(s) + \gamma(1-R)V_t(s_{t+1})). \tag{5}$$

Compared to the model-based approach, our approach is model-free. It does not require the prior knowledge of the uncertainty set, i.e., the knowledge of $p_s^a, \forall s \in \mathcal{S}, a \in \mathcal{A}$. Furthermore, the memory requirement of our algorithm is $|\mathcal{S}| \times |\mathcal{A}|$ (used to store the Q-table), and unlike the model-based approach it does not need a table of size $|\mathcal{S}|^2|\mathcal{A}|$ to store $p_s^a, \forall s \in \mathcal{S}, a \in \mathcal{A}$, which could be problematic if the state space is large. Moreover, our algorithm does not involve a relaxation of the uncertainty set like the one in [Roy et al., 2017], which will incur a stringent condition on the discount factor to guarantee the convergence. As will be shown below, the convergence of our Algorithm 1 does not require any condition on the discount factor.

We show in the following theorem that the robust Q-learning algorithm converges asymptotically to the optimal robust action-value function Q^*.

Theorem 2. *(Asymptotic Convergence) If step sizes α_t satisfy that $\sum_{t=0}^{\infty} \alpha_t = \infty$ and $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$, then $Q_t \to Q^*$ as $t \to \infty$ with probability 1.*

To further establish the finite-time error bound for our robust Q-learning algorithm in Algorithm 1, we make the following assumption that is commonly used in the analysis of vanilla Q-learning.

Assumption 1. *The Markov chain induced by the behavior policy π_b and the transition kernel $p_s^a, \forall s \in \mathcal{S}, a \in \mathcal{A}$ is uniformly ergodic.*

Let μ_{π_b} denote the stationary distribution over $\mathcal{S} \times \mathcal{A}$ induced by π_b and $p_s^a, \forall s \in \mathcal{S}, a \in \mathcal{A}$. We then further define $\mu_{\min} = \min_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mu_{\pi_b}(s,a)$. This quantity characterizes how many samples are needed to visit every state-action pair sufficiently often. Define the following mixing time of the induced Markov chain: $t_{\text{mix}} = \min\left\{t : \max_{s\in\mathcal{S}} d_{\text{TV}}(\mu_\pi, P(s_t = \cdot|s_0 = s)) \leq \frac{1}{4}\right\}$, where d_{TV} is the total variation distance.

The following theorem establishes the finite-time error bound of our robust Q-learning algorithm.

Theorem 3. *(Finite-Time Error Bound) There exist some positive constants c_0 and c_1 such that for any $\delta < 1$, any $\epsilon < \frac{1}{1-\gamma}$, any T satisfying*

$$T \geq c_0 \left(\frac{1}{\mu_{min}(1-\gamma)^5\epsilon^2} + \frac{t_{mix}}{\mu_{min}(1-\gamma)} \right) \log\left(\frac{T|\mathcal{S}||\mathcal{A}|}{\delta} \right) \log\left(\frac{1}{\epsilon(1-\gamma)^2} \right), \tag{6}$$

and step size $\alpha_t = \frac{c_1}{\log\left(\frac{T|\mathcal{S}||\mathcal{A}|}{\delta}\right)} \min\left(\frac{1}{t_{mix}}, \frac{\epsilon^2(1-\gamma)^4}{\gamma^2} \right), \forall t \geq 0$ we have with probability at least $1-6\delta$, $\|Q_T - Q^*\|_\infty \leq 3\epsilon$.

From the theorem, we can see that to guarantee an ϵ-accurate estimate, a sample size $\tilde{\mathcal{O}}(\frac{1}{\mu_{\min}(1-\gamma)^5\epsilon^2} + \frac{t_{\text{mix}}}{\mu_{\min}(1-\gamma)})$ (up to some logarithmic terms) is needed. This complexity matches with the one for the vanilla Q-learning in [Li et al., 2020] (within a constant factor), while our algorithm also guarantees robustness to MDP model uncertainty. Our algorithm design and analysis can be readily extended to robustify TD and SARSA. The variance-reduction technique [Wainwright, 2019] can also be combined with our robust Q-learning algorithm to further improve the dependency on $(1-\gamma)$.

5 Function Approximation: Robust TDC

In this section, we investigate the case where the state and action spaces can be large or even continuous. A popular approach is to approximate the value function using a parameterized function, e.g., linear function and neural network. In this section, we focus on the case with linear function approximation to illustrate the main idea of designing robust RL algorithms. Our approach can be extended to non-linear (smooth) function approximation using techniques in, e.g., [Cai et al., 2019, Bhatnagar et al., 2009, Wai et al., 2019, Wang et al., 2021].

We focus on the problem of robust policy evaluation, i.e., estimate the robust value function V^π defined in (1) for a given policy π under the worst-case MDP transition kernel in the uncertainty set. Note that for robust RL with model uncertainty, any policy evaluation problem can be viewed as "off-transition-kernel", as it is to evaluate the value function under the worst-case MDP using samples from a different MDP. Since the TD algorithm with function approximation may diverge under off-policy training [Baird, 1995] and importance sampling cannot be applied here due to unknown transition kernel, in this paper we generalize the GTD method [Maei et al., 2010, Maei, 2011] to the robust setting.

Let $\left\{\phi^{(i)} : \mathcal{S} \to \mathbb{R},\, i = 1, \ldots, N\right\}$ be a set of N fixed base functions, where $N \ll |\mathcal{S}||\mathcal{A}|$. In particular, we approximate the robust value function using a linear combination of $\phi^{(i)}$'s: $V_\theta(s) = \sum_{i=1}^N \theta^i \phi_s^{(i)} = \phi_s^\top \theta$, where $\theta \in \mathbb{R}^N$ is the weight vector.

Define the following robust Bellman operator for a given policy π:

$$\begin{aligned}\mathbf{T}_\pi V(s) &\triangleq \mathbb{E}_{A\sim\pi(\cdot|s)}[c(s,A) + \gamma\sigma_{\mathcal{P}_s^A}(V)] \\ &= \mathbb{E}_{A\sim\pi(\cdot|s)}\left[c(s,A) + \gamma(1-R)\sum_{s'\in\mathcal{S}} p_{s,s'}^A V(s') + \gamma R \max_{s'\in\mathcal{S}} V(s') \right].\end{aligned} \tag{7}$$

We then define the mean squared projected robust Bellman error (MSPRBE) as

$$\text{MSPRBE}(\theta) = \|\mathbf{\Pi}\mathbf{T}_\pi V_\theta - V_\theta\|_{\mu_\pi}^2, \tag{8}$$

where $\|v\|_{\mu_\pi}^2 = \int v^2(s)\mu_\pi(ds)$, μ_π is the stationary distribution induced by π, and $\mathbf{\Pi}$ is a projection onto the linear function space w.r.t. $\|\cdot\|_{\mu_\pi}$. We will develop a two time-scale gradient-based approach to minimize the MSPRBE. However, it can be seen that $\max_s V_\theta(s)$ in (7) is not smooth in θ, which is troublesome in both algorithm design and analysis. To solve this issue, we introduce the following smoothed robust Bellman operator $\hat{\mathbf{T}}_\pi$ by smoothing the max with a LSE(LogSumExp):

$$\hat{\mathbf{T}}_\pi V(s) = \mathbb{E}_{A\sim\pi(\cdot|s)}\left[c(s,A) + \gamma(1-R)\sum_{s'\in\mathcal{S}} p_{s,s'}^A V(s') + \gamma R \cdot \text{LSE}(V) \right], \tag{9}$$

where $\text{LSE}(V) = \frac{\log\left(\sum_s e^{\varrho V(s)}\right)}{\varrho}$ is the LogSumExp w.r.t. V with a parameter $\varrho > 0$. Note that when $\varrho \to \infty$, the smoothed robust Bellman operator $\hat{\mathbf{T}}_\pi \to \mathbf{T}_\pi$. The LSE operator can also be replaced by some other operator that approximates the max operator and is smooth, e.g., mellow-max [Asadi and Littman, 2017]. In the following, we first show that the fixed point of $\hat{\mathbf{T}}_\pi$ exists for any ϱ, and the fixed points converge to the one of $\mathbf{T}_\pi$ for large ϱ.

Theorem 4. *(1). For any ϱ, $\hat{\mathbf{T}}_\pi$ has a fixed point.*

(2). Let V_1 and V_2 be the fixed points of $\hat{\mathbf{T}}_\pi$ and $\mathbf{T}_\pi$, respectively. Then

$$\|V_1 - V_2\|_\infty \leq \frac{\gamma R}{1-\gamma}\frac{\log|\mathcal{S}|}{\varrho} \to 0, \text{ as } \varrho \to \infty. \tag{10}$$

We then denote by $J(\theta)$ the smoothed MSPRBE with the LSE operator, and the goal is:

$$\min_\theta J(\theta) = \min_\theta \left\|\mathbf{\Pi}\hat{\mathbf{T}}_\pi V_\theta - V_\theta\right\|^2_{\mu_\pi}. \tag{11}$$

5.1 Algorithm Development

In the following, we develop the robust TDC algorithm to solve the problem in (11). We will first derive the gradient of the smoothed MSPRBE, $J(\theta)$, and then design a two time-scale update rule using the weight doubling trick in [Sutton et al., 2009] to solve the double sampling problem. Define $\delta_{s,a,s'}(\theta) \triangleq c(s,a) + \gamma(1-R)V_\theta(s') + \gamma R\text{LSE}(V_\theta) - V_\theta(s)$, where $\text{LSE}(V_\theta)$ is the LogSumExp function w.r.t. $V_\theta = \theta^\top\phi$. Denote by $C \triangleq \mathbb{E}_{\mu_\pi}\left[\phi_S^\top\phi_S\right]$. Then, $\mathbb{E}_\mu[\delta_{S,A,S'}(\theta)\phi_S] = \Phi^\top D\left(\hat{\mathbf{T}}_\pi V_\theta - V_\theta\right)$, where $D = \text{diag}(\mu_\pi(s_1), \mu_\pi(s_2), ..., \mu_\pi(s_{|\mathcal{S}|}))$ and $\Phi = (\phi_{s_1}, \phi_{s_2}, ..., \phi_{s_{|\mathcal{S}|}})^\top \in \mathbb{R}^{|\mathcal{S}|\times N}$. We know that $\mathbf{\Pi}^\top D\mathbf{\Pi} = D^\top\Phi(\Phi^\top D\Phi)^{-1}\Phi^\top D$ from [Maei, 2011]. Hence we have

$$J(\theta) = \left\|\mathbf{\Pi}\hat{\mathbf{T}}_\pi V_\theta - V_\theta\right\|^2_{\mu_\pi} = \mathbb{E}_{\mu_\pi}[\delta_{S,A,S'}(\theta)\phi_S]^\top C^{-1}\mathbb{E}_{\mu_\pi}[\delta_{S,A,S'}(\theta)\phi_S]. \tag{12}$$

Then, its gradient can be written as:

$$\begin{aligned}
&-\frac{1}{2}\nabla J(\theta) = -\mathbb{E}_{\mu_\pi}[(\nabla\delta_{S,A,S'}(\theta))\phi_S]^\top C^{-1}\mathbb{E}_{\mu_\pi}[\delta_{S,A,S'}(\theta)\phi_S] \\
&= \mathbb{E}_{\mu_\pi}[\delta_{S,A,S'}(\theta)\phi_S] - \gamma\mathbb{E}_{\mu_\pi}\left[\left((1-R)\phi_{S'} + R\cdot\nabla\text{LSE}(V_\theta)\right)\phi_S^\top\right]\omega(\theta),
\end{aligned}$$

where $\omega(\theta) = C^{-1}\mathbb{E}_{\mu_\pi}[\delta_{S,A,S'}(\theta)\phi_S]$. It can be seen that to obtain an unbiased estimate of $\nabla J(\theta)$, two independent samples are needed as there exists a multiplication of two expectations, which is not applicable when there is only one sample trajectory. We then utilize the weight doubling trick in [Sutton et al., 2009], and design the robust TDC algorithm in Algorithm 2. Specifically, we introduce a fast time scale to estimate $\omega(\theta)$, and a slow time scale to estimate $\nabla J(\theta)$. Denote the projection by $\mathbf{\Pi}_K(x) \triangleq \arg\min_{\|y\|\leq K}\|y - x\|$ for any $x \in \mathbb{R}^N$. Our robust TDC algorithm in Algorithm 2 can be implemented in an online and incremental fashion. If the uncertainty set becomes a singleton, i.e., $R = 0$, then Algorithm 2 reduces to the vanilla TDC algorithm.

5.2 Finite-Time Error Bound of Robust TDC

Unlike the vanilla TDC algorithm, $J(\theta)$ here is non-convex. Therefore, we are interested in the convergence to stationary points, i.e., the rate of $\|\nabla J(\theta)\| \to 0$. We first make some standard assumptions which are commonly used in RL algorithm analysis, e.g., [Wang and Zou, 2020, Kaledin et al., 2020, Xu et al., 2019, Srikant and Ying, 2019, Bhandari et al., 2018].

Assumption 2 (Bounded feature)**.** $\|\phi_s\|_2 \leq 1, \forall s \in \mathcal{S}$.

Assumption 3 (Bounded cost function)**.** $|c(s,a)| \leq c_{\max}, \forall s \in \mathcal{S}$ *and* $a \in \mathcal{A}$.

Assumption 4 (Problem solvability)**.** *The matrix $C = \mathbb{E}_{\mu_\pi}[\phi_S\phi_S^\top]$ is non-singular with $\lambda > 0$ being its smallest eigenvalue.*

Algorithm 2 Robust TDC with Linear Function Approximation

Input: T,α, β, ϱ, ϕ_i for $i = 1, ..., N$, projection radius K
Initialization: θ_0,w_0, s_0

Choose $W \sim \text{Uniform}(0, 1, ..., T-1)$
for $t = 0, 1, 2, ..., W-1$ **do**
  Take action according to $\pi(\cdot|s_t)$ and observe s_{t+1} and c_t
  $\phi_t \leftarrow \phi_{s_t}$
  $\delta_t(\theta_t) \leftarrow c_t + \gamma(1-R)V_{\theta_t}(s_{t+1}) + \gamma R \frac{\log(\sum_s e^{\varrho \theta^\top \phi_s})}{\varrho} - V_{\theta_t}(s_t)$
  $\theta_{t+1} \leftarrow \mathbf{\Pi}_K \left(\theta_t + \alpha \left(\delta_t(\theta_t)\phi_t - \gamma \left((1-R)\phi_{t+1} + R \sum_{s \in \mathcal{S}} \left(\frac{e^{\varrho V_\theta(s)} \phi_s}{\sum_{j \in \mathcal{S}} e^{\varrho V_\theta(j)}} \right) \right) \phi_t^\top \omega_t \right) \right)$
  $\omega_{t+1} \leftarrow \mathbf{\Pi}_K(\omega_t + \beta(\delta_t(\theta_t) - \phi_t^\top \omega_t)\phi_t)$
end for

Output: θ_W

Assumption 5 (Geometric uniform ergodicity)**.** *There exist some constants $m > 0$ and $\rho \in (0, 1)$ such that for any $t > 0$, $\max_{s \in \mathcal{S}} d_{TV}(\mathbb{P}(s_t|s_0 = s), \mu_\pi) \leq m\rho^t$.*

In the following theorem, we characterize the finite-time error bound for the convergence of our robust TDC algorithm. Here we only provide the order of the bounds in terms of T. The explicit bounds can be found in (129) in Appendix D.3.

Theorem 5. *Consider the following step-sizes: $\beta = \mathcal{O}\left(\frac{1}{T^b}\right)$, and $\alpha = \mathcal{O}\left(\frac{1}{T^a}\right)$, where $\frac{1}{2} < a \leq 1$ and $0 < b \leq a$. Then we have that*

$$\mathbb{E}[\|\nabla J(\theta_W)\|^2] = \mathcal{O}\left(\frac{1}{T\alpha} + \alpha \log(1/\alpha) + \frac{1}{T\beta} + \beta \log(1/\beta)\right), \tag{13}$$

If we further let $a = b = 0.5$, then $\mathbb{E}[\|\nabla J(\theta_W)\|^2] = \mathcal{O}\left(\frac{\log T}{\sqrt{T}}\right)$.

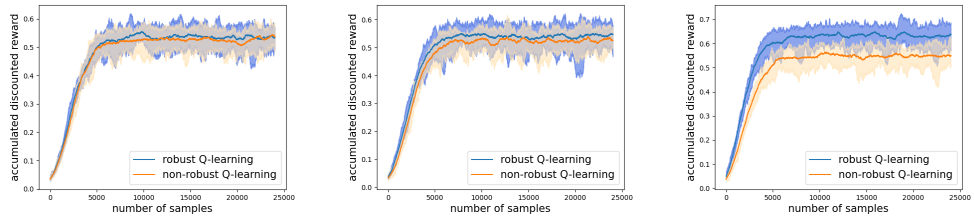
The robust TDC has a matching complexity with the vanilla TDC with non-linear function approximation [Wang et al., 2021], but provides the additional robustness to model uncertainty. It does not need to relax the uncertainty set like in [Roy et al., 2017], and our convergence results do not need a condition on the discount factor.

6 Experiments

6.1 Robust Q-Learning

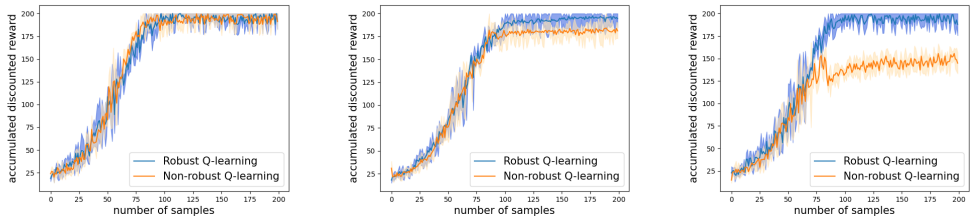
In this section, we compare our robust Q-learning with the vanilla non-robust Q-learning. We use OpenAI gym framework [Brockman et al., 2016], and consider two different problems: Frozen lake and Cart-Pole. One more example of the taxi problem is given in the appendix. To demonstrate the robustness, the policy is learned in a perturbed MDP, and is then tested on the true unperturbed MDP. Specifically, during the training, we set a probability p such that after the agent takes an action, with probability p, the state transition is uniformly over $\mathcal{S}$, and with probability $1-p$ the state transition is according to the true unperturbed transition kernel. The behavior policy for all the experiments below is set to be a uniform distribution over the action space given any state, i.e., $\pi_b(a|s) = \frac{1}{|\mathcal{A}|}$ for any $s \in \mathcal{S}$ and $a \in \mathcal{A}$. We then evaluate the performance of the obtained policy in the unperturbed environment. At each time t, the policy we evaluate is the greedy-policy w.r.t. the current estimate of the Q-function, i.e., $\pi_t(s) = \arg\max_a Q_t(s, a)$. A Monte-Carlo method with horizon 100 is used to evaluate the accumulated discounted reward of the learned policy on the unperturbed MDP. We take the average over 30 trajectories. More details are provided in the appendix.

In Figure 1 and Figure 2, we plot the accumulated discounted reward of both algorithms under different p and R for both problems. The upper and lower envelopes of the curves correspond to the 95 and 5 percentiles of the 30 trajectories, respectively. It can be seen that overall our robust Q-learning algorithm achieves a much higher reward than the vanilla Q-learning. This demonstrates the robustness of our robust Q-learning algorithm to model uncertainty. Moreover, as p and R getting



(a) p=0.1, R=0.1 (b) p=0.05, R=0.2 (c) p=0.1, R=0.2

Figure 1: **FrozenLake-v0**: robust Q-learning v.s. non-robust Q-learning.



(a) p=0.1, R=0.1 (b) p=0.05, R=0.2 (c) p=0.1, R=0.2

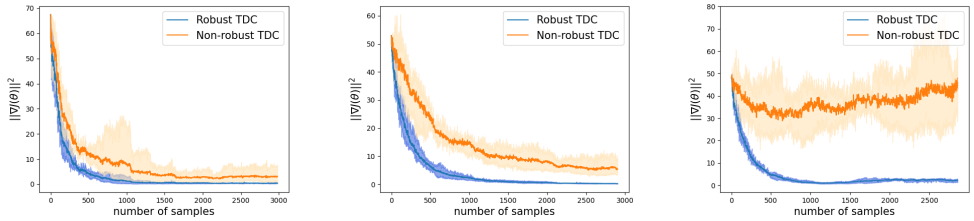
Figure 2: **CartPole-v0**: robust Q-learning v.s. non-robust Q-learning.

larger, i.e., as the MDP that we learn the policy deviates from the MDP we test the policy, the advantage of our robust Q-learning algorithm is getting more significant compared to the vanilla Q-learning algorithm.

6.2 Robust TDC with Linear Function Approximation

In this section we compare our robust TDC with the vanilla non-robust TDC with linear function approximation on the 4×4 Frozen Lake problem. The problem setting is the same as the one in Section 6.1. More details about the experiment setup are provided in the appendix.

We implement the two algorithms using samples from the perturbed MDP both for 30 times, and obtain 30 sequences of $\{\theta_t^i\}_{t=1}^{\infty}$, $i = 1, 2, ..., 30$. We then compute the squared gradient norm $\|\nabla J(\theta)\|^2$ on the true unperturbed MDP, and see whether $\{\theta_t^i\}_{t=1}^{\infty}$ converges to some stationary points on the true unperturbed MDP. In Fig. 3, we plot the average squared gradient norm $\|\nabla J(\theta)\|^2$ for different p and R. The upper and lower envelops are the 95 and 5 percentiles of the 30 curves. It can be seen that our robust TDC converges much faster than vanilla TDC, and as the model mismatch between the training and test MDPs enlarges, the vanilla TDC may diverge (Fig. 3(c)), while our robust TDC still converges to some stationary point. Also, the robust TDC has a much smaller variance, which indicates a much stable behavior under model uncertainty.



(a) p=0.1, R=0.1 (b) p=0.05, R=0.2 (c) p=0.1, R=0.2

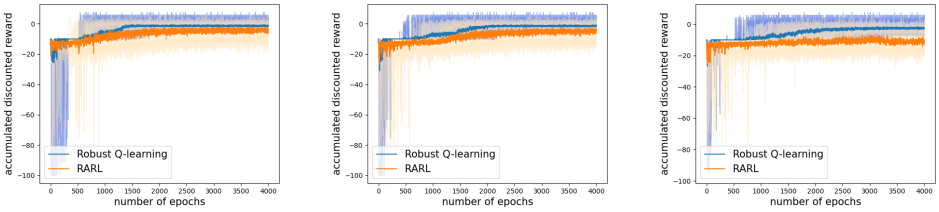
Figure 3: **FrozenLake-v0**: $\|\nabla J(\theta)\|^2$ of robust TDC and non-robust TDC.

6.3 Comparison with The Adversarial Training Approach

We also compare our robust Q-learning with Robust Adversarial Reinforcement Learning (RARL) in [Pinto et al., 2017]. To apply their algorithm to our problem setting, we model the nature as an adversarial player, and its goal is to minimize the reward that the agent receives. The action space $\mathcal{A}_{ad}$ of the nature is set to be the state space $\mathcal{A}_{ad} \triangleq \mathcal{S}$. Then the perturbed training environment can be viewed as an adversarial model: both the agent and the adversary take actions a_a, a_{ad}, then the environment will transit to state a_{ad} with probability R and transit following the unperturbed MDP $p_s^{a_a}$ with probability $1 - R$. The goal of the maximize its accumulated reward, while the goal of the natural is to minimize it.

Following the RARL algorithm [Pinto et al., 2017], in each iteration of the training, we first fix the adversarial policy and use Q-learning to optimize the agent's policy and obtain the Q-table Q_t. Then we fix the agent's policy and optimize the adversarial policy.

After each training iteration, we test the performance of the greedy policies w.r.t. Q-tables obtained from robust Q-learning and RARL. The testing environment is set to be the worst-case, i.e., after the agent takes an action, the environment transits to the state which has the minimal value function ($\arg\min_{s\in\mathcal{S}} V_t(s)$) with probability p. We plot the accumulated discounted rewards of both algorithms against number of training iterations under different parameters. We set $\alpha = 0.2$ and $\gamma = 0.9$. It can be seen from Fig. 4 that our robust Q-learning achieves a higher accumulative reward, and thus is more robust that the RARL algorithm in [Pinto et al., 2017]. Also our robust Q-learning is more stable during training, i.e., the variance is smaller.



(a) p=0.1, R=0.1 (b) p=0.05, R=0.2 (c) p=0.1, R=0.2

Figure 4: **Taxi-v3**: robust Q-learning v.s. RARL.

7 Conclusion

In this paper, we develop a novel approach for solving model-free robust RL problems with model uncertainty. Our algorithms can be implemented in an online and incremental fashion, do not require additional memory than their non-robust counterparts. We theoretically proved the convergence of our algorithms under no additional assumption on the discount factor, and further characterized their finite-time error bounds, which match with their non-robust counterparts (within a constant factor). Our approach can be readily extended to robustify TD, SARSA and other GTD algorithms. **Limitations:** It is also of future interest to investigate robustness to reward uncertainty, and other types of uncertainty sets, e.g., ones defined by KL divergence, Wasserstein distance and total variation. **Negative societal impact:** To the best of the authors' knowledge, this study does not have any potential negative impact on the society.

8 Acknowledgment

The work of Y. Wang and S. Zou was supported by the National Science Foundation under Grants CCF-2106560 and CCF- 2007783.

References

Mohammed Amin Abdullah, Hang Ren, Haitham Bou Ammar, Vladimir Milenkovic, Rui Luo, Mingtian Zhang, and Jun Wang. Wasserstein robust reinforcement learning. *arXiv preprint arXiv:1907.13196*, 2019.

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.

Kavosh Asadi and Michael L Littman. An alternative softmax operator for reinforcement learning. In *Proc. International Conference on Machine Learning (ICML)*, volume 70, pages 243–252. JMLR, 2017.

Christopher G Atkeson and Jun Morimoto. Nonparametric representation of policies and value functions: A trajectory-based approach. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 1643–1650, 2003.

author=Dalal, Gal and Szörényi, Balázs and Thoppe, Gugan and Mannor, Shie. Finite sample analyses for TD(0) with function approximation. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, pages 6144–6160, 2018.

Kishan Panaganti Badrinath and Dileep Kalathil. Robust reinforcement learning using least squares policy iteration with provable performance guarantees. In *Proc. International Conference on Machine Learning (ICML)*, pages 511–520. PMLR, 2021.

J Andrew Bagnell, Andrew Y Ng, and Jeff G Schneider. Solving uncertain markov decision. 09 2001.

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings*, pages 30–37. Elsevier, 1995.

Carolyn L Beck and Rayadurgam Srikant. Error bounds for constant step-size Q-learning. *Systems & control letters*, 61(12):1203–1208, 2012.

Jalaj Bhandari and Daniel Russo. Global optimality guarantees for policy gradient methods. *arXiv preprint arXiv:1906.01786*, 2019.

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

Shalabh Bhatnagar, Doina Precup, David Silver, Richard S Sutton, Hamid Maei, and Csaba Szepesvári. Convergent temporal-difference learning with arbitrary smooth function approximation. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, volume 22, pages 1204–1212, 2009.

Vivek S Borkar and Sean P Meyn. The ode method for convergence of stochastic approximation and reinforcement learning. *SIAM Journal on Control and Optimization*, 38(2):447–469, 2000.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

Qi Cai, Zhuoran Yang, Jason D Lee, and Zhaoran Wang. Neural temporal-difference learning converges to global optima. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 11312–11322, 2019.

Shicong Cen, Chen Cheng, Yuxin Chen, Yuting Wei, and Yuejie Chi. Fast global convergence of natural policy gradient methods with entropy regularization. *arXiv preprint arXiv:2007.06558*, 2020.

Zaiwei Chen, Sheng Zhang, Thinh T Doan, Siva Theja Maguluri, and John-Paul Clarke. Performance of Q-learning with linear function approuimation: Stability and finite-time analysis. *arXiv preprint arXiv:1905.11425*, 2019.

Gal Dalal, Balázs Szörényi, Gugan Thoppe, and Shie Mannor. Finite sample analysis of two-timescale stochastic approximation with applications to reinforcement learning. *Proceedings of Machine Learning Research*, 75:1–35, 2018.

Gal Dalal, Balázs Szörényi, and Gugan Thoppe. A tale of two-timescale reinforcement learning with the tightest finite-time bound. In *Proc. Conference on Artificial Intelligence (AAAI)*, pages 3701–3708, 2020.

Thinh T Doan. Finite-time analysis and restarting scheme for linear two-time-scale stochastic approximation. *SIAM Journal on Control and Optimization*, 59(4):2798–2819, 2021.

Eyal Even-Dar, Yishay Mansour, and Peter Bartlett. Learning rates for Q-learning. *Journal of machine learning Research*, 5(1), 2003.

Harsh Gupta, R Srikant, and Lei Ying. Finite-time performance bounds and adaptive learning rate selection for two time-scale reinforcement learning. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 4706–4715, 2019.

Linfang Hou, Liang Pang, Xin Hong, Yanyan Lan, Zhiming Ma, and Dawei Yin. Robust reinforcement learning with wasserstein constraint. *arXiv preprint arXiv:2006.00945*, 2020.

Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. In *Proc. International Conference on Learning Representations (ICLR)*, 2017.

P. J. Huber. A robust version of the probability ratio test. *Ann. Math. Statist.*, 36:1753–1758, 1965.

Garud N Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2): 257–280, 2005.

Maxim Kaledin, Eric Moulines, Alexey Naumov, Vladislav Tadic, and Hoi-To Wai. Finite time analysis of linear two-timescale stochastic approximation with Markovian noise. In *Proc. Annual Conference on Learning Theory (CoLT)*, pages 2144–2203. PMLR, 2020.

Jernej Kos and Dawn Song. Delving into adversarial attacks on deep policies. In *Proc. International Conference on Learning Representations (ICLR)*, 2017.

Harshat Kumar, Alec Koppel, and Alejandro Ribeiro. On the sample complexity of actor-critic method for reinforcement learning with function approximation. *arXiv preprint arXiv:1910.08412*, 2019.

Gen Li, Yuting Wei, Yuejie Chi, Yuantao Gu, and Yuxin Chen. Sample complexity of asynchronous Q-learning: Sharper analysis and variance reduction. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Gen Li, Changxiao Cai, Yuxin Chen, Yuantao Gu, Yuting Wei, and Yuejie Chi. Is Q-learning minimax optimal? a tight sample complexity analysis. *arXiv preprint arXiv:2102.06548*, 2021.

Shiau Hong Lim, Huan Xu, and Shie Mannor. Reinforcement learning in robust markov decision processes. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 701–709, 2013.

Yen-Chen Lin, Zhang-Wei Hong, Yuan-Hong Liao, Meng-Li Shih, Ming-Yu Liu, and Min Sun. Tactics of adversarial attack on deep reinforcement learning agents. In *Proc. International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 3756–3762, 2017.

Bo Liu, Ji Liu, Mohammad Ghavamzadeh, Sridhar Mahadevan, and Marek Petrik. Finite-sample analysis of proximal gradient td algorithms. In *Proc. International Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 504–513. Citeseer, 2015.

Shaocong Ma, Yi Zhou, and Shaofeng Zou. Variance-reduced off-policy TDC learning: Non-asymptotic convergence analysis. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 14796–14806, 2020.

Shaocong Ma, Yi Zhou, and Shaofeng Zou. Greedy-GQ with variance reduction: Finite-time analysis and improved complexity. In *Proc. International Conference on Learning Representations (ICLR)*, 2021.

Hamid Reza Maei. Gradient temporal-difference learning algorithms. *Thesis, University of Alberta*, 2011.

Hamid Reza Maei, Csaba Szepesvári, Shalabh Bhatnagar, and Richard S Sutton. Toward off-policy learning control with function approximation. In *Proc. International Conference on Machine Learning (ICML)*, pages 719–726, 2010.

Ajay Mandlekar, Yuke Zhu, Animesh Garg, Li Fei-Fei, and Silvio Savarese. Adversarially robust policy learning: Active construction of physically-plausible perturbations. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3932–3939. IEEE, 2017.

Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *Proc. International Conference on Machine Learning (ICML)*, pages 6820–6829. PMLR, 2020.

Jun Morimoto and Kenji Doya. Robust reinforcement learning. *Neural computation*, 17(2):335–359, 2005.

Arnab Nilim and Laurent El Ghaoui. Robustness in Markov decision problems with uncertain transition matrices. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 839–846, 2004.

Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommannan, and Girish Chowdhary. Robust deep reinforcement learning with adversarial attacks. In *Proc. International Conference on Autonomous Agents and MultiAgent Systems*, pages 2040–2042, 2018.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *Proc. International Conference on Machine Learning (ICML)*, pages 2817–2826. PMLR, 2017.

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

Guannan Qu and Adam Wierman. Finite-time analysis of asynchronous stochastic approximation and Q-learning. In *Proc. Annual Conference on Learning Theory (CoLT)*, pages 3185–3205. PMLR, 2020.

Aravind Rajeswaran, Sarvjeet Ghotra, Balaraman Ravindran, and Sergey Levine. Epopt: Learning robust neural network policies using model ensembles. In *Proc. International Conference on Learning Representations (ICLR)*, 2017.

Aurko Roy, Huan Xu, and Sebastian Pokutta. Reinforcement learning under model mismatch. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 3046–3055, 2017.

Jay K Satia and Roy E Lave Jr. Markovian decision processes with uncertain transition probabilities. *Operations Research*, 21(3):728–740, 1973.

R. Srikant and Lei Ying. Finite-time error bounds for linear stochastic approximation and TD learning. In *Proc. Annual Conference on Learning Theory (CoLT)*, pages 2803–2830, 2019.

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

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction, Second Edition*. The MIT Press, Cambridge, Massachusetts, 2018.

Richard S Sutton, Csaba Szepesvári, and Hamid Reza Maei. A convergent O(n) algorithm for off-policy temporal-difference learning with linear function approximation. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, volume 21, pages 1609–1616. MIT Press, 2008.

Richard S Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proc. International Conference on Machine Learning (ICML)*, pages 993–1000, 2009.

Aviv Tamar, Shie Mannor, and Huan Xu. Scaling up robust mdps using function approximation. In *Proc. International Conference on Machine Learning (ICML)*, pages 181–189. PMLR, 2014.

Eugene Vinitsky, Yuqing Du, Kanaad Parvate, Kathy Jang, Pieter Abbeel, and Alexandre Bayen. Robust reinforcement learning using adversarial populations. *arXiv preprint arXiv:2008.01825*, 2020.

Hoi-To Wai, Mingyi Hong, Zhuoran Yang, Zhaoran Wang, and Kexin Tang. Variance reduced policy evaluation with smooth function approximation. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pages 5784–5795, 2019.

Martin J Wainwright. Variance-reduced Q-learning is minimax optimal. *arXiv preprint arXiv:1906.04697*, 2019.

Lingxiao Wang, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Neural policy gradient methods: Global optimality and rates of convergence. In *Proc. International Conference on Learning Representations (ICLR)*, 2020.

Yue Wang and Shaofeng Zou. Finite-sample analysis of Greedy-GQ with linear function approximation under Markovian noise. In *Proc. International Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 11–20. PMLR, 2020.

Yue Wang, Shaofeng Zou, and Yi Zhou. Finite-sample analysis for two time-scale non-linear TDC with general smooth function approximation. *arXiv preprint arXiv:2104.02836*, 2021.

Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust Markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013.

Yue Wu, Weitong Zhang, Pan Xu, and Quanquan Gu. A finite time analysis of two time-scale actor critic methods. *arXiv preprint arXiv:2005.01350*, 2020.

Tengyu Xu and Yingbin Liang. Sample complexity bounds for two timescale value-based reinforcement learning algorithms. In *Proc. International Conference on Artifical Intelligence and Statistics (AISTATS)*, pages 811–819. PMLR, 2021.

Tengyu Xu, Shaofeng Zou, and Yingbin Liang. Two time-scale off-policy TD learning: Non-asymptotic analysis over Markovian samples. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 10633–10643, 2019.

Zhuoran Yang, Yongxin Chen, Mingyi Hong, and Zhaoran Wang. Provably global convergence of actor-critic: A case for linear quadratic regulator with ergodic cost. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 8353–8365, 2019.

Kaiqing Zhang, Bin Hu, and Tamer Basar. On the stability and convergence of robust adversarial reinforcement learning: A case study on linear quadratic systems. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020a.

Kaiqing Zhang, Tao Sun, Yunzhe Tao, Sahika Genc, Sunil Mallya, and Tamer Basar. Robust multi-agent reinforcement learning with model uncertainty. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020b.

Shaofeng Zou, Tengyu Xu, and Yingbin Liang. Finite-sample analysis for SARSA with linear function approximation. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 8665–8675, 2019.

Supplementary Materials

A Proof of Theorem 2: Asymptotic Convergence of Robust Q-Learning

In this section we show that the robust Q-learning converges exactly to the optimal robust Q function Q^*. Recall that the optimal robust Q function Q^* is the solution to the robust Bellman operator $\mathbf{T}$:

$$Q^*(s,a) = c(s,a) + \gamma \sigma_{\mathcal{P}_s^a}((\min_{a\in\mathcal{A}} Q^*(s_1,a), \min_{a\in\mathcal{A}} Q^*(s_2,a), ..., \min_{a\in\mathcal{A}} Q^*(s_{|\mathbb{S}|},a))^\top). \tag{14}$$

It can be shown that the estimated update is an unbiased estimation of $\mathbf{T}$. More specifically,

$$\begin{aligned}
\mathbf{T}Q(s,a) &= c(s,a) + \gamma \sigma_{\mathcal{P}_s^a}(V) \\
&= c(s,a) + \gamma(1-R)(p_s^a)^\top V + R\max_{s'} V(s') \\
&= c(s,a) + \gamma(1-R)\sum_{s'}(p_{s,s'}^a)V(s') + R\max_{s'} V(s') \\
&= c(s,a) + \gamma \sum_{s'} p_{s,s'}^a \left((1-R)(\mathbb{1}_{s'})^\top V + R\max_q q^\top V\right),
\end{aligned} \tag{15}$$

which is the expectation of the estimated update in line 5 of Algorithm 1.

A.1 Robust Bellman operator is a contraction

It was shown in [Iyengar, 2005, Roy et al., 2017] that the robust Bellman operator is a contraction. Here, for completeness, we include the proof for our R-contamination uncertainty set. More specifically,

$$\begin{aligned}
&|\mathbf{T}Q(s,a) - \mathbf{T}Q'(s,a)| \\
&= |c(s,a) + \gamma\sigma_{\mathcal{P}_s^a}(V) - c(s,a) - \gamma\sigma_{\mathcal{P}_s^a}(V')| \\
&= \gamma|\sigma_{\mathcal{P}_s^a}(V) - \sigma_{\mathcal{P}_s^a}(V')| \\
&= \gamma|\max_q \left\{(1-R)(p_s^a)^\top V + Rq^\top V\right\} - \max_{q'}\left\{(1-R)(p_s^a)^\top V' + Rq'^\top V'\right\}| \\
&= \gamma\left|\sum_{s'\in\mathbb{S}} p_{s,s'}^a\left((1-R)V(s')\right) + R\max_{s'} V(s') - \sum_{s'\in\mathbb{S}} p_{s,s'}^a\left((1-R)V'(s')\right) - R\max_{s'} V'(s')\right| \\
&= \gamma\left|\sum_{s'\in\mathbb{S}} p_{s,s'}^a(1-R)\left(V(s') - V'(s')\right) + R(\max_{s'} V(s') - \max_{s'} V'(s'))\right| \\
&\le \gamma\left|\sum_{s'\in\mathbb{S}} p_{s,s'}^a(1-R)\left(\min_a Q(s',a) - \min_b Q'(s',b)\right)\right| + \gamma R(|\max_{s'} V(s') - \max_{s'} V'(s')|) \\
&\le \gamma\sum_{s'\in\mathbb{S}} p_{s,s'}^a(1-R)\left|\left(\min_a Q(s',a) - \min_b Q'(s',b)\right)\right| + \gamma R\max_s|(V(s) - V'(s))| \\
&\overset{(a)}{\le} \gamma\sum_{s'\in\mathbb{S}} p_{s,s'}^a(1-R)\|Q-Q'\|_\infty + \gamma R\|Q-Q'\|_\infty \\
&\le \gamma\|Q-Q'\|_\infty,
\end{aligned} \tag{16}$$

where (a) can be shown as below. Assume that $a_1 = \arg\min_a Q(s',a)$ and $b_1 = \arg\min_a Q'(s',a)$. Then if $Q(s',a_1) > Q'(s',b_1)$, then

$$|Q(s',a_1) - Q'(s',b_1)| = Q(s',a_1) - Q'(s',b_1) \le Q(s',b_1) - Q'(s',b_1) \le \|Q-Q'\|_\infty. \tag{17}$$

Similarly, it can also be shown when $Q(s',a_1) \le Q'(s',b_1)$, and hence the inequality (a) holds.

A.2 Asymptotic Convergence of Robust Q-Leaning

With the definition of $\mathbf{T}$, the update (5) of robust Q-learning can be re-written as a stochastic approximation:

$$Q_{t+1}(s_t, a_t) = (1-\alpha_t)Q_t(s_t, a_t) + \alpha_t(\mathbf{T}Q_t(s_t, a_t) + \eta_t(s_t, a_t, s_{t+1})), \tag{18}$$

where the noise term is

$$\eta_t(s_t, a_t, s_{t+1}) = c(s_t, a_t) + \gamma R \max_s V_t(s) + \gamma(1-R)V_t(s_{t+1}) - \mathbf{T}Q_t(s_t, a_t). \tag{19}$$

From (15), we have that

$$\mathbb{E}[\eta_t(S_t, A_t, S_{t+1})|S_t = s_t, A_t = a_t] = 0. \tag{20}$$

The variance can be bounded by

$$\mathbb{E}[(\eta_t(S_t, A_t, S_{t+1}))^2] \leq \gamma^2(1-R)^2(\max_{s,a} Q_t^2(s,a)), \tag{21}$$

where the last inequality is from $V_t(s_{t+1}) \leq \max_s V_t(s) \leq \max_{s,a} Q_t(s,a)$. Thus the noise term η_t has zero mean and bounded variance. From [Borkar and Meyn, 2000], we know that the stochastic approximation (18) converges to the fixed point of $\mathbf{T}$, i.e., Q^*. Hence we showed that robust Q-learning converges to optimal optimal robust Q function Q^* with probability 1.

B Finite-Time Analysis of Robust Q-Learning

In this section, we develop the finite-time analysis of the Algorithm 1.

B.1 Notations

We first introduce some notations. For a vector $v = (v_1, v_2, ..., v_n)$, we denote the entry wise absolute value $(|v_1|, ..., |v_n|)$ by $|v|$. For a sample $O_t = (s_t, a_t, s_{t+1})$, define $\Lambda_{t+1} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times|\mathcal{S}||\mathcal{A}|}$ as

$$\Lambda_{t+1}((s,a),(s',a')) = \begin{cases} \alpha, & \text{if } (s,a) = (s',a') = (s_t, a_t), \\ 0, & \text{otherwise.} \end{cases} \tag{22}$$

Also we define the sample transition matrix $P_{t+1} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times|\mathcal{S}|}$ as

$$P_{t+1}((s,a),s') = \begin{cases} 1, & \text{if } (s,a,s') = O_t, \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

We also define the transition kernel matrix $P \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times|\mathcal{S}|}$ as

$$P((s,a),s') = p^a_{s,s'}. \tag{24}$$

We use $Q_t \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ and $V_t \in \mathbb{R}^{|\mathcal{S}|}$ to denote the vectors of value functions. Denote the cost function $c \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ with entry $c(s,a)$ being the cost received at (s,a). Then the update of robust Q-learning (5) can be written in matrix form as

$$Q_t = (I - \Lambda_t)Q_{t-1} + \Lambda_t\Big(c + \gamma(1-R)P_tV_{t-1} + \gamma R \max_{s\in\mathcal{S}} V_{t-1}(s)P_t\mathbf{1}\Big), \tag{25}$$

where $\mathbf{1}$ denotes the vector $(1,1,1,...,1)^\top \in \mathbb{R}^{|\mathcal{S}|}$. The robust Bellman equation can be written as

$$Q^* = c + \gamma(1-R)PV^* + \gamma R \max_{s\in\mathcal{S}} V^*(s)P\mathbf{1}. \tag{26}$$

B.2 Analysis

Define $\psi_t = Q_t - Q^*$, then by (25) and (26), we have that

$$\begin{aligned}\psi_t &= Q_t - Q^* \\ &= (I-\Lambda_t)Q_{t-1} + \Lambda_t(c + \gamma(1-R)P_tV_{t-1} + \gamma R \max_{s\in\mathcal{S}} V_{t-1}(s)P_t\mathbf{1}) - Q^*\end{aligned}$$

$$
\begin{aligned}
&= (I - \Lambda_t)(Q_{t-1} - Q^*) + \Lambda_t(c + \gamma(1-R)P_tV_{t-1} + \gamma R \max_{s\in\mathcal{S}} V_{t-1}(s)P_t\mathbf{1} - Q^*) \\
&= (I - \Lambda_t)\psi_{t-1} + \Lambda_t(\gamma(1-R)P_tV_{t-1} + \gamma R \max_{s\in\mathcal{S}} V_{t-1}(s)P_t\mathbf{1} - \gamma(1-R)PV^* \\
&\quad - \gamma R \max_{s\in\mathcal{S}} V^*(s)P\mathbf{1}) \\
&= (I - \Lambda_t)\psi_{t-1} + \gamma(1-R)\Lambda_t \underbrace{(P_tV_{t-1} - PV^*)}_{k_1} \\
&\quad + \gamma R\Lambda_t \underbrace{(\max_{s\in\mathcal{S}} V_{t-1}(s)P_t\mathbf{1} - \max_{s\in\mathcal{S}} V^*(s)P\mathbf{1}))}_{k_2}. \qquad (27)
\end{aligned}
$$

The term k_1 can be written as

$$
P_tV_{t-1} - PV^* = P_tV_{t-1} - P_tV^* + P_tV^* - PV^* = P_t(V_{t-1} - V^*) + (P_t - P)V^*. \qquad (28)
$$

Similarly, we have that

$$
k_2 = \Big(\max_{s\in\mathcal{S}} V_{t-1}(s) - \max_{s\in\mathcal{S}} V^*(s)\Big)P_t\mathbf{1} + \max_{s\in\mathcal{S}} V^*(s)(P_t - P)\mathbf{1}. \qquad (29)
$$

Hence (27) can be written as

$$
\begin{aligned}
\psi_t &= Q_t - Q^* \\
&= (I - \Lambda_t)\psi_{t-1} + \gamma(1-R)\Lambda_t(P_t(V_{t-1} - V^*) + (P_t - P)V^*) \\
&\quad + \gamma R\Lambda_t\left(\left(\max_{s\in\mathcal{S}} V_{t-1}(s) - \max_{s\in\mathcal{S}} V^*(s)\right)P_t\mathbf{1} + \max_{s\in\mathcal{S}} V^*(s)(P_t - P)\mathbf{1}\right) \\
&= (I - \Lambda_t)\psi_{t-1} + \left(\gamma(1-R)\Lambda_t(P_t - P)V^*) + \gamma R\Lambda_t(\max_{s\in\mathcal{S}} V^*(s)(P_t - P)\mathbf{1})\right) \\
&\quad + \left(\gamma(1-R)\Lambda_t(P_t(V_{t-1} - V^*)) + \gamma R\Lambda_t\left(\left(\max_{s\in\mathcal{S}} V_{t-1}(s) - \max_{s\in\mathcal{S}} V^*(s)\right)P_t\mathbf{1}\right)\right). \qquad (30)
\end{aligned}
$$

By applying (30) recursively, we have that

$$
\begin{aligned}
\psi_t = &\underbrace{\prod_{j=1}^{t}(I - \Lambda_j)\psi_0}_{k_{1,t}} \\
&+ \underbrace{\gamma(1-R)\sum_{i=1}^{t}\prod_{j=i+1}^{t}(I - \Lambda_j)\Lambda_i(P_i - P)V^* + \gamma R\sum_{i=1}^{t}\prod_{j=i+1}^{t}(I - \Lambda_j)\Lambda_i \max_{s\in\mathcal{S}} V^*(s)(P_i - P)\mathbf{1}}_{k_{2,t}} \\
&+ \underbrace{\gamma(1-R)\sum_{i=1}^{t}\prod_{j=i+1}^{t}(I - \Lambda_j)\Lambda_iP_i(V_{i-1} - V^*) + \gamma R\sum_{i=1}^{t}\prod_{j=i+1}^{t}(I - \Lambda_j)\Lambda_i(\max_{s\in\mathcal{S}} V_{i-1}(s) - \max_{s\in\mathcal{S}} V^*(s))P_i\mathbf{1}}_{k_{3,t}}.
\end{aligned} \qquad (31)
$$

We then bound terms $k_{i,t}$ separately.

Lemma 1. *Define* $t_{frame} = \frac{443t_{mix}}{\mu_{\min}}\log\frac{4|\mathcal{S}||\mathcal{A}|T}{\delta}$. *Then with probability at least* $1-\delta$, *for any* $(s,a) \in \mathcal{S}\times\mathcal{A}$ *and any* $t \geq t_{frame}$, $k_{1,t}$ *can be bounded as*

$$
|k_{1,t}| \leq (1-\alpha)^{\frac{t\mu_{\min}}{2}}\|\psi_0\|_\infty\mathbf{1}; \qquad (32)
$$

and for $t < t_{frame}$,

$$
|k_{1,t}| \leq \|\psi_0\|_\infty\mathbf{1}. \qquad (33)
$$

Proof. First note that the (s,a)-entry of $k_{1,t}$ can be written as

$$
k_{1,t}(s,a) = (1-\alpha)^{K_t(s,a)}\psi_0(s,a), \qquad (34)
$$

where $K_t(s,a)$ denotes the times that the sample trajectory visits (s,a) before the time step t. We introduce a lemma from [Li et al., 2020] first:

Lemma 2. *(Lemma 5 [Li et al., 2020]) For a time-homogeneous and uniformly ergodic Markov chain with state space $\mathcal{X}$ and any $0 < \delta < 1$, if $t \geq \frac{443 t_{mix}}{\mu_{\min}} \log \frac{|\mathcal{X}|}{\delta}$, then for any $y \in \mathcal{X}$,*

$$\mathbb{P}_{X_1=y}\left\{\exists x \in \mathcal{X} : \sum_{j=1}^{t} \mathbb{1} X_j = x \leq \frac{t\mu(x)}{2}\right\} \leq \delta, \tag{35}$$

where $t_{mix} = \min\left\{t : \max_{x\in\mathcal{X}} d_{TV}(\mu, P^t(\cdot|x)) \leq \frac{1}{4}\right\}$; μ is the stationary distribution of the Markov chain, and $\mu_{\min} \triangleq \min_{x\in\mathcal{X}} \mu(x)$.

From this lemma, we know that for any $(s,a) \in \mathcal{S}\times\mathcal{A}$ and any $t \geq \frac{443 t_{\text{mix}}}{\mu_{\min}} \log \frac{4|\mathcal{S}||\mathcal{A}|T}{\delta}$, we have that

$$K_t(s,a) \geq \frac{t\mu_{\min}}{2}, \tag{36}$$

with probability at least $1-\delta$.

Thus (34) can be bounded as

$$|k_{1,t}(s,a)| \leq (1-\alpha)^{\frac{t\mu_{\min}}{2}} |\psi_0(s,a)| \tag{37}$$

with probability at least $1-\delta$ for any $(s,a) \in \mathcal{S}\times\mathcal{A}$ and any $t \geq \frac{443 t_{\text{mix}}}{\mu_{\min}} \log \frac{4|\mathcal{S}||\mathcal{A}|T}{\delta}$, which shows the claim.

For $t < t_{\text{frame}}$, the bound is obvious by noting that $\|I - \Lambda_j\| \leq 1$. □

Lemma 3. *There exists some constant $\hat{c}$, such that for any $\delta < 1$ and any $t \leq T$ that satisfies $0 < \alpha \log \frac{|\mathcal{S}||\mathcal{A}|T}{\delta} < 1$, with probability at least $1 - \frac{\delta}{|\mathcal{S}||\mathcal{A}|T}$,*

$$|k_{2,t}| \leq 5\gamma\hat{c}\sqrt{\alpha \log \frac{T|\mathcal{S}||\mathcal{A}|}{\delta}} \|V^*(s)\|_\infty \mathbf{1}, \tag{38}$$

Proof. Recall that

$$k_{2,t} = \gamma(1-R)\sum_{i=1}^{t}\prod_{j=i+1}^{t}(I-\Lambda_j)\Lambda_i(P_i - P)V^* + \gamma R\sum_{i=1}^{t}\prod_{j=i+1}^{t}(I-\Lambda_j)\Lambda_i(P_i-P)w^*, \tag{39}$$

where $w^* \triangleq \max_{s\in\mathcal{S}} V^*(s)\mathbf{1}$. Then the (s,a)-th entry of $k_{2,t}$ can be written as

$$\begin{aligned} k_{2,t}(s,a) = \gamma(1-R)&\sum_{i=1}^{K_t(s,a)} \alpha(1-\alpha)^{K_t(s,a)-i}(P_{t_i+1}(s,a) - P(s,a))V^* \\ &+ \gamma R \sum_{i=1}^{K_t(s,a)} \alpha(1-\alpha)^{K_t(s,a)-i}(P_{t_i+1}(s,a) - P(s,a))w^*, \end{aligned} \tag{40}$$

where $t_i(s,a)$ is the time step when the trajectory visits (s,a) for the i-th time. We define $\text{Var}_P(V) \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ being a vector, where $\text{Var}_P(V)(s,a) = \sum_{s'\in\mathcal{S}} p^a_{s,s'}(V(s')^2) - (\sum_{s'\in\mathcal{S}} p^a_{s,s'}V(s'))^2 \triangleq \text{Var}_{P^a_s}[V]$ for any $V \in \mathbb{R}^{|\mathcal{S}|}$.

From Section E.1 in [Li et al., 2020], we know that

$$\text{Var}\left[\sum_{i=1}^{K} \alpha(1-\alpha)^{K-i}(P_{t_i+1}(s,a) - P(s,a))V^*\right] = \alpha\text{Var}_{P^a_s}[V^*] \triangleq \sigma_K^2 \tag{41}$$

for some constant σ_K^2 and any $K \leq T$. Moreover, note that

$$\text{Var}\left[\sum_{i=1}^{K} \alpha(1-\alpha)^{K-i}(P_{t_i+1}(s,a) - P(s,a))w^*\right]$$

$$\overset{(a)}{=} \sum_{i=1}^{K} \alpha^2(1-\alpha)^{2K-2i}\text{Var}[(P_{t_i+1}(s,a) - P(s,a))w^*]$$
$$\overset{(b)}{=} \sum_{i=1}^{K} \alpha^2(1-\alpha)^{2K-2i}\text{Var}[\max_s V^*(s)((P_{t_i+1}(s,a) - P(s,a))\mathbf{1})]$$
$$= 0, \tag{42}$$

where equation (a) is due to the fact that $\{P_{t_1+1}(s,a), P_{t_2+1}(s,a), ..., P_{t_i+1}(s,a)\}_{i\in\mathbb{N}}$ are independent (Equation (101) in [Li et al., 2020]), (b) is from the definition of ω^*, and the last equation is because the sum of each entries of $P_{t_i+1}(s,a) - P(s,a)$ is 0.

the last equality is due to the fact that every entries of w^* are the same and hence $\text{Var}_{P_s^a}[w^*] = 0$.

Additionally, we have that

$$\left\|\alpha(1-\alpha)^{K-i}(P_{t_i+1}(s,a) - P(s,a))V^*\right\|_\infty \leq 2\alpha\|V^*(s)\|_\infty \triangleq D, \tag{43}$$

where we denote the bound by D. Also,

$$\left\|\alpha(1-\alpha)^{K-i}(P_{t_i+1}(s,a) - P(s,a))w^*\right\|_\infty \leq D. \tag{44}$$

Hence from the Bernstein inequality ([Li et al., 2020]), we have that

$$\begin{aligned}
&|k_{2,t}(s,a)| \\
&\leq \gamma(1-R)\hat{c}\left(\sqrt{\sigma_K^2 \log\left(\frac{T|\mathcal{S}||\mathcal{A}|}{\delta}\right)} + D\log\frac{T|\mathcal{S}||\mathcal{A}|}{\delta}\right) + \gamma R\hat{c}\left(D\log\frac{T|\mathcal{S}||\mathcal{A}|}{\delta}\right) \\
&\leq 5\gamma\hat{c}\sqrt{\alpha\log\frac{T|\mathcal{S}||\mathcal{A}|}{\delta}}\|V^*(s)\|_\infty,
\end{aligned} \tag{45}$$

for some constant $\hat{c}$ with probability at least $1 - \frac{\delta}{|\mathcal{S}||\mathcal{A}|T}$, and the last step is due to the fact that $\text{Var}_{P_s^a}[V^*] \leq \|V^*\|_\infty^2$ and $\alpha\log\frac{|\mathcal{S}||\mathcal{A}|T}{\delta} < 1$. This hence completes the proof. □

Lemma 4. *For any $t \geq T$,*

$$|k_{3,t}| \leq \gamma\sum_{i=1}^{t}\|\psi_{i-1}\|_\infty \prod_{j=i+1}^{t}(I - \Lambda_j)(\Lambda_i)\mathbf{1}. \tag{46}$$

Proof. First note that for any i,

$$\|P_i(V_{i-1} - V^*)\|_\infty \leq \|P_i\|_1\|V_{i-1} - V^*\|_\infty = \|V_{i-1} - V^*\|_\infty \leq \|\psi_{i-1}\|_\infty, \tag{47}$$

where the last inequality is from

$$\begin{aligned}
&\|V_{i-1} - V^*\|_\infty = \max_s |V_{i-1}(s) - V^*(s)| = |V_{i-1}(s^*) - V^*(s^*)| \\
&= |\min_a Q_{i-1}(s^*,a) - \min_b Q^*(s^*,b)| \leq \|Q_{i-1} - Q^*\|_\infty,
\end{aligned} \tag{48}$$

where $s^* = \arg\max|V_{i-1}(s) - V^*(s)|$. Similarly,

$$\left\|(\max_{s\in\mathcal{S}} V_{i-1}(s) - \max_{s\in\mathcal{S}} V^*(s))P_i\mathbf{1}\right\|_\infty \leq |\max_{s\in\mathcal{S}} V_{i-1}(s) - \max_{s\in\mathcal{S}} V^*(s)| \leq \|\psi_{i-1}\|_\infty, \tag{49}$$

where the last inequality is from $|\max_{s\in\mathcal{S}} V_{i-1}(s) - \max_{s\in\mathcal{S}} V^*(s)| \leq \|V_{i-1} - V^*\|_\infty \leq \|Q_{i-1} - Q^*\|_\infty$. Hence $K_{3,t}$ can be bounded as

$$|k_{3,t}| \leq \gamma\sum_{i=1}^{t}\|\psi_{i-1}\|_\infty \prod_{j=i+1}^{t}(I - \Lambda_j)(\Lambda_i)\mathbf{1}. \tag{50}$$

□

Now combine the bounds for terms $k_{1,t}, k_{2,t}$ and $k_{3,t}$, we have the bound on ψ_t as follows.

For $t < t_{\text{frame}}$, we have that

$$\|\psi_t\|_\infty \leq \|\psi_0\|_\infty \mathbf{1} + 5\gamma\hat{c}\sqrt{\alpha \log \frac{T|\mathcal{S}||\mathcal{A}|}{\delta}}\|V^*(s)\|_\infty \mathbf{1} + \gamma \sum_{i=1}^{t} \|\psi_{i-1}\|_\infty \prod_{j=i+1}^{t} (I - \Lambda_j)(\Lambda_i)\mathbf{1}; \tag{51}$$

and for $t \geq t_{\text{frame}}$, we have that

$$\|\psi_t\|_\infty \leq (1-\alpha)^{\frac{t\mu_{\min}}{2}} \|\psi_0\|_\infty \mathbf{1} + 5\gamma\hat{c}\sqrt{\alpha \log \frac{T|\mathcal{S}||\mathcal{A}|}{\delta}}\|V^*(s)\|_\infty \mathbf{1} + \gamma \sum_{i=1}^{t} \|\psi_{i-1}\|_\infty \prod_{j=i+1}^{t} (I - \Lambda_j)(\Lambda_i)\mathbf{1}. \tag{52}$$

This bound exactly matches the bound in Equation (42) in [Li et al., 2020] and hence the remaining proof for Theorem 3 can be obtained by following the proof in [Li et al., 2020]. We omit the remaining proof and only state the result.

Theorem 6. *Define*

$$t_{th} = \max\left\{\frac{2\log\frac{1}{(1-\gamma)^2\epsilon}}{\alpha\mu_{\min}}, t_{frame}\right\}; \tag{53}$$

$$\mu_{frame} = \frac{1}{2}\mu_{\min} t_{frame}; \tag{54}$$

$$\rho = (1-\gamma)(1-(1-\alpha)^{\mu_{frame}}), \tag{55}$$

then for any $\delta < 1$ and any $\epsilon < \frac{1}{1-\gamma}$, there exists a universal constant $\hat{c}$ and c_0 (determined by $\hat{c}$), such that with probability at least $1 - 6\delta$, the following bound holds for any $t < T$:

$$\|Q_t - Q^*\|_\infty \leq \frac{(1-\rho)^k \|Q_0 - Q^*\|_\infty}{1-\gamma} + \frac{5\hat{c}\gamma}{1-\gamma}\sqrt{\alpha \log \frac{|\mathcal{S}||\mathcal{A}|T}{\delta}} + \epsilon, \tag{56}$$

where $k = \max\left\{0, \lfloor \frac{t-t_{th}}{t_{frame}} \rfloor\right\}$, as long as

$$T \geq c_0 \left(\frac{1}{\mu_{min}(1-\gamma)^5\epsilon^2} + \frac{t_{mix}}{\mu_{min}(1-\gamma)}\right)\log\left(\frac{T|\mathcal{S}||\mathcal{A}|}{\delta}\right)\log\left(\frac{1}{\epsilon(1-\gamma)^2}\right),$$

and step size $0 < \alpha \log\left(\frac{|\mathcal{S}||\mathcal{A}|T}{\delta}\right) < 1$.

This theorem implies that the convergence rate of our robust Q-learning is as fast as the one of the vanilla Q-learning algorithm in [Li et al., 2020](except the constant $\hat{c}$).

Finally, to show Theorem 3, we only need to show each term in (56) is smaller than ϵ. It can be verified that there exists constants c_1, such that if we choose the step size $\alpha = \frac{c_1}{\log\left(\frac{T|\mathcal{S}||\mathcal{A}|}{\delta}\right)} \min\left(\frac{1}{t_{\text{mix}}}, \frac{\epsilon^2(1-\gamma)^4}{\gamma^2}\right)$, then $\frac{(1-\rho)^k\|Q_0-Q^*\|_\infty}{1-\gamma} \leq \epsilon$ (inequality (51) in [Li et al., 2020]) and $\frac{5\hat{c}\gamma}{1-\gamma}\sqrt{\alpha \log \frac{|\mathcal{S}||\mathcal{A}|T}{\delta}} \leq \epsilon$ (by choosing suitable constant c_1). Then we have that $\|Q_t - Q^*\|_\infty \leq 3\epsilon$. This completes the proof.

C Proof of Theorem 4: Approximation of Smoothing Robust Bellman Operator

In this section we prove Theorem 4. First note that for any $x, y \in \mathbb{R}^{|\mathcal{S}|}$,

$$|\text{LSE}(x) - \text{LSE}(y)| \leq \sup_{t\in[0,1]} \|\nabla \text{LSE}(tx + (1-t)y)\|_1 \|x - y\|_\infty. \tag{57}$$

It can be shown that the gradient of LSE is softmax, i.e.,

$$\frac{\partial \text{LSE}(x)}{\partial x_i} = \frac{e^{\varrho x_i}}{\sum_j e^{\varrho x_j}}. \tag{58}$$

Hence

$$\|\nabla \text{LSE}(z)\|_1 = 1, \forall z \in \mathbb{R}^{|\mathcal{S}|}, \tag{59}$$

which implies that $|\text{LSE}(x) - \text{LSE}(y)| \leq \|x - y\|_\infty$. Hence for any $x, y \in \mathbb{R}^{|\mathcal{S}|}$, we have that

$$\begin{aligned}
|\hat{\mathbf{T}}_\pi x(s) - \hat{\mathbf{T}}_\pi y(s)| &= \left| \mathbb{E}_A \left[\gamma(1-R) \sum_{s' \in \mathcal{S}} p^A_{s,s'}(x(s') - y(s')) + \gamma R(\text{LSE}(x) - \text{LSE}(y)) \right] \right| \\
&\leq \gamma(1-R)\|x - y\|_\infty + \gamma R \|x - y\|_\infty \\
&\leq \gamma \|x - y\|_\infty.
\end{aligned} \tag{60}$$

This means that $\hat{\mathbf{T}}_\pi$ is a contraction, which implies that it has a fixed point.

We then show the limit of the fixed points of $\hat{\mathbf{T}}_\pi$ is the fixed point of $\mathbf{T}_\pi$ Note that $\mathbf{T}_\pi V_1 = V_1$ and $\hat{\mathbf{T}}_\pi V_2 = V_2$, hence

$$\begin{aligned}
&\|V_1 - V_2\|_\infty \\
&= \|\mathbf{T}_\pi V_1 - \hat{\mathbf{T}}_\pi V_2\|_\infty \\
&\leq \|\mathbf{T}_\pi V_1 - \mathbf{T}_\pi V_2\|_\infty + \|\mathbf{T}_\pi V_2 - \hat{\mathbf{T}}_\pi V_2\|_\infty \\
&= \max_s \left| \mathbb{E}_\pi \Big[\gamma (1-R) \sum_{s'} p^A_{s,s'} V_1(s') + \gamma R \max_{s'} V_1(s') \right. \\
&\quad \left. - \gamma (1-R) \sum_{s'} p^A_{s,s'} V_2(s') - \gamma R \max_{s'} V_2(s') \Big] \right| \\
&\quad + \max_s \left| \mathbb{E}_\pi \left[\gamma R \left(\max_{s'} V_2(s') - \text{LSE}(V_2) \right) \right] \right| \\
&\leq \max_s \mathbb{E}_\pi \left[\left| \gamma (1-R) \sum_{s'} p^A_{s,s'} (V_1(s') - V_2(s')) \right| + \left| \gamma R \left(\max_{s'} V_1(s') - \max_{s'} V_2(s') \right) \right| \right] \\
&\quad + \max_s \left| \mathbb{E}_\pi \left[\gamma R \left(\max_{s'} V_2(s') - \text{LSE}(V_2) \right) \right] \right| \\
&\overset{(a)}{\leq} \max_s \gamma |V_1(s) - V_2(s)| + \left| \mathbb{E}_\pi \left[\gamma R \left(\max_{s'} V_2(s') - \text{LSE}(V_2) \right) \right] \right| \\
&\leq \gamma \|V_1 - V_2\|_\infty + \gamma R \frac{\log |\mathcal{S}|}{\varrho},
\end{aligned} \tag{61}$$

where (a) is from $|V_1(s') - V_2(s')| \leq \max_s |V_1(s) - V_2(s)| = \|V_1 - V_2\|_\infty$ and $|\max_{s'} V_1(s') - \max_{s'} V_2(s')| \leq \|V_1 - V_2\|_\infty$, and the last inequality is from $\text{LSE}(V) - \max V = \frac{\log(\sum_s e^{\varrho V(s)}) - \log e^{\varrho \max V}}{\varrho} = \frac{1}{\varrho} \log \frac{\sum_s e^{\varrho V(s)}}{e^{\varrho \max V}} = \frac{1}{\varrho} \log \sum_s e^{\varrho V(s) - \varrho \max V} \leq \frac{\log |\mathcal{S}|}{\varrho}$. Hence this completes the proof.

D Proof of Theorem 5: Finite-Time Analysis of Robust TDC with Linear Function Approximation

In this section we develop the finite-time analysis of the robust TDC algorithm. In the following proofs, $\|v\|$ denotes the l_2 norm if v is a vector; and $\|A\|$ denotes the operator norm if A is a matrix.

For the convenience of proof, we add a projection step to the algorithm, i.e., we let

$$\theta_{t+1} \leftarrow \mathbf{\Pi}_K \left(\theta_t + \alpha \left(\delta_t(\theta_t)\phi_t - \gamma \left((1-R)\phi_{t+1} + R \sum_{s \in \mathcal{S}} \left(\frac{e^{\varrho V_\theta(s)} \phi_s}{\sum_{j \in \mathcal{S}} e^{\varrho V_\theta(j)}} \right) \right) \phi_t^\top \omega_t \right) \right),$$

$$\omega_{t+1} \leftarrow \mathbf{\Pi}_K \left(\omega_t + \beta(\delta_t(\theta_t) - \phi_t^\top \omega_t)\phi_t\right), \tag{62}$$

for some constant K. We note that recently there are several works [Srikant and Ying, 2019, Xu and Liang, 2021, Kaledin et al., 2020] on finite-time analysis of RL algorithms that do not need the projection. However, a direct generalization of their approach does not necessarily work in our case. Specifically, the problem in [Srikant and Ying, 2019] is for one time scale *linear* stochastic approximation. and doesn't need to consider the effect of the ω_t introduced, also their work highly depends on the bound of the update functions of θ_t (see inequality (18) in [Srikant and Ying, 2019]). The parameter θ_t in [Srikant and Ying, 2019] is bounded using itself at a previous timestep by taking advantage of the fact that the update of θ is linear. However, in our problem, the update is not linear in θ, and our update rule is two time-scale. The approach in [Kaledin et al., 2020] transforms the original two time-scale updates into two asymptotically independent updates via a linear mapping, which is however challenging for our non-linear updates. Some other work, e.g., [Xu and Liang, 2021], gets around this issue by imposing additional assumptions on the function class. Specifically, it is assumed that V_θ (non-linear function approximation) is bounded for all θ. For the linear function approximation setting considered in this paper, this assumption is equivalent to the assumption of a finite θ, which is guaranteed by the projection step in this paper.

D.1 Lipschitz Smoothness

In this section, we first show that $\nabla J(\theta)$ is Lipschitz. We begin with an important lemma.

Lemma 5. *For any $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, both $\delta_{s,a,s'}(\theta)$ and $\nabla\delta_{s,a,s'}(\theta)$ are bounded and Lipschitz, i.e., for any θ and θ',*

$$|\delta_{s,a,s'}(\theta)| \le c_{\max} + \gamma R(K + \frac{\log|\mathcal{S}|}{\varrho}) + (1+\gamma)K \triangleq C_\delta, \tag{63}$$

$$\|\delta_{s,a,s'}(\theta) - \delta_{s,a,s'}(\theta')\| \le (1+\gamma)\|\theta - \theta'\| \triangleq L_\delta\|\theta - \theta'\|, \tag{64}$$

$$\|\nabla\delta_{s,a,s'}(\theta) - \nabla\delta_{s,a,s'}(\theta')\| \le 2\gamma R\varrho\|\theta - \theta'\| \triangleq L'_\delta\|\theta - \theta'\|. \tag{65}$$

Proof. **1. δ is bounded:**

Recall that

$$\delta_{s,a,s'}(\theta) = c(s,a) + \gamma(1-R)V_\theta(s') + \gamma R\frac{\log(\sum_{j\in\mathcal{S}} e^{\varrho\theta^\top\phi_j})}{\varrho} - V_\theta(s). \tag{66}$$

First we have that

$$\begin{aligned}|\delta_{s,a,s'}(\theta)| &\le c_{\max} + \gamma(1-R)K + \gamma R\frac{\log|\mathcal{S}|e^{K\varrho}}{\varrho} + \gamma RK + K \\ &= c_{\max} + \gamma R(K + \frac{\log|\mathcal{S}|}{\varrho}) + (1+\gamma)K.\end{aligned} \tag{67}$$

2. δ is Lipschitz:

The Lipschitz smoothness of $\delta_{s,a,s'}$ can be showed by finding the bound of $\nabla\delta_{s,a,s'}$. We first recall that

$$\nabla\delta_{s,a,s'}(\theta) = \gamma(1-R)\phi_{s'} + \gamma R\frac{\sum_i e^{\varrho\theta^\top\phi_i}\phi_i}{\sum_j e^{\varrho\theta^\top\phi_j}} - \phi_s. \tag{68}$$

Hence

$$\|\nabla\delta_{s,a,s'}(\theta)\| \le \gamma(1-R) + 1 + \gamma R = 1 + \gamma. \tag{69}$$

3. $\nabla\delta$ is Lipschitz:

Finally we need to verify the Lipschitz smoothness of $\nabla\delta_{s,a,s'}(\theta)$, which can be implied from the bound of $\nabla^2\delta_{s,a,s'}(\theta)$. First we have that

$$\nabla^2\delta_{s,a,s'}(\theta) = \gamma R\varrho\frac{\sum_{i,j} e^{\varrho\theta^\top(\phi_i+\phi_j)}\phi_i^\top\phi_i - \sum_{i,j} e^{\varrho\theta^\top(\phi_i+\phi_j)}\phi_i^\top\phi_j}{(\sum_j e^{\varrho\theta^\top\phi_j})^2} \le 2\gamma R\varrho. \tag{70}$$

□

With this lemma, we then show that $\nabla J(\theta)$ is Lipschitz as follows.

Lemma 6. *For any θ and θ', we have that*

$$\|\nabla J(\theta) - \nabla J(\theta')\| \leq 2\left(\frac{L_\delta^2}{\lambda} + \frac{C_\delta L'_\delta}{\lambda}\right)\|\theta - \theta'\| \triangleq L_J\|\theta - \theta'\|. \tag{71}$$

Proof. From Lemma 5, we have that

$$\|\mathbb{E}_{\mu_\pi}[(\nabla\delta_{S,A,S'}(\theta))\phi_S]\| \leq L_\delta \tag{72}$$

and

$$\|\mathbb{E}_{\mu_\pi}[(\nabla\delta_{S,A,S'}(\theta))\phi_S] - \mathbb{E}_{\mu_\pi}[(\nabla\delta_{S,A,S'}(\theta'))\phi_S]\| \leq L'_\delta\|\theta - \theta'\|. \tag{73}$$

Also it is easy to see that

$$\|C^{-1}\mathbb{E}_{\mu_\pi}[\delta_{S,A,S'}(\theta)\phi_S]\| \leq \frac{1}{\lambda}C_\delta, \tag{74}$$

and

$$\|C^{-1}\mathbb{E}_{\mu_\pi}[\delta_{S,A,S'}(\theta)\phi_S] - C^{-1}\mathbb{E}_{\mu_\pi}[\delta_{S,A,S'}(\theta')\phi_S]\| \leq \frac{1}{\lambda}L_\delta\|\theta - \theta'\|. \tag{75}$$

Thus this implies that

$$\|\nabla J(\theta) - \nabla J(\theta')\| \leq 2\left(\frac{L_\delta^2}{\lambda} + \frac{C_\delta L'_\delta}{\lambda}\right)\|\theta - \theta'\|, \tag{76}$$

and hence completes the proof. □

D.2 Tracking Error

In this section, we study the bound of the tracking error, which is defined as $z_t = \omega_t - \omega(\theta_t)$. First we can rewrite the fast time-scale update in Algorithm 1 as follows:

$$\begin{aligned}
z_{t+1} &= \omega_{t+1} - \omega(\theta_{t+1}) \\
&= \omega_t + \beta(\delta_t(\theta_t) - \phi_t^\top\omega_t)\phi_t - \omega(\theta_{t+1}) \\
&= z_t + \omega(\theta_t) + \beta(\delta_t(\theta_t) - \phi_t^\top\omega_t)\phi_t - \omega(\theta_{t+1}) \\
&= z_t + \omega(\theta_t) + \beta(\delta_t(\theta_t) - \phi_t^\top(z_t + \omega(\theta_t)))\phi_t - \omega(\theta_{t+1}) \\
&= z_t + \omega(\theta_t) + \beta\delta_t(\theta_t)\phi_t - \beta\phi_t^\top z_t\phi_t - \beta\phi_t^\top\omega(\theta_t)\phi_t - \omega(\theta_{t+1}) \\
&= z_t - \beta\phi_t\phi_t^\top z_t + \beta(\delta_t(\theta_t)\phi_t - \phi_t\phi_t^\top\omega(\theta_t)) + \omega(\theta_t) - \omega(\theta_{t+1}).
\end{aligned} \tag{77}$$

Thus taking the norm of both sides implies that

$$\begin{aligned}
\|z_{t+1}\|^2 &\overset{(a)}{\leq} \|z_t\|^2 + 3\beta^2\|z_t\|^2 + 3\beta^2\|\delta_t(\theta_t)\phi_t - \phi_t\phi_t^\top\omega(\theta_t)\|^2 + 3\|\omega(\theta_t) - \omega(\theta_{t+1})\|^2 \\
&\quad + 2\langle z_t, -\beta\phi_t\phi_t^\top z_t\rangle + 2\langle z_t, \beta(\delta_t(\theta_t)\phi_t - \phi_t\phi_t^\top\omega(\theta_t))\rangle + 2\langle z_t, \omega(\theta_t) - \omega(\theta_{t+1})\rangle \\
&= \|z_t\|^2 - 2\beta z_t^\top C z_t + 3\beta^2\|z_t\|^2 + 3\beta^2\|\delta_t(\theta_t)\phi_t - \phi_t\phi_t^\top\omega(\theta_t)\|^2 + 3\|\omega(\theta_t) - \omega(\theta_{t+1})\|^2 \\
&\quad + 2\beta\langle z_t, (C - \phi_t\phi_t^\top)z_t\rangle + 2\langle z_t, \beta(\delta_t(\theta_t)\phi_t - \phi_t\phi_t^\top\omega(\theta_t))\rangle + 2\langle z_t, \omega(\theta_t) - \omega(\theta_{t+1})\rangle \\
&\overset{(b)}{\leq} (1 + 3\beta^2 - 2\beta\lambda)\|z_t\|^2 + \beta^2 C_1 + 2\beta\langle z_t, (C - \phi_t\phi_t^\top)z_t\rangle + 2\langle z_t, \omega(\theta_t) - \omega(\theta_{t+1})\rangle \\
&\quad + 2\langle z_t, \beta(\delta_t(\theta_t)\phi_t - \phi_t\phi_t^\top\omega(\theta_t))\rangle,
\end{aligned} \tag{78}$$

where(a) is from $\|x + y + z\|^2 \leq 3\|x\|^2 + 3\|y\|^2 + 3\|z\|^2$ for any $x, y, z \in \mathbb{R}^N$, (b) is from $z_t^\top C z_t \geq \lambda\|z_t\|^2$, and $C_1 = 3\left(C_\delta + \frac{C_\delta}{\lambda}\right)^2 + 3\left(C_\delta + (1 + 2R\varrho K)\frac{C_\delta}{\lambda}\right)^2$ is the upper bound of $3\|\delta_t(\theta_t)\phi_t - \phi_t\phi_t^\top\omega(\theta_t)\|^2 + \frac{3}{\beta^2}\|\omega(\theta_t) - \omega(\theta_{t+1})\|^2$.

Taking expectation on both sides and applying recursively (78), we obtain that

$$\mathbb{E}[\|z_{t+1}\|^2] \leq q^{t+1}\|z_0\|^2 + 2\sum_{j=0}^{t} q^{t-j}\beta\mathbb{E}[f(z_j, O_j)] + 2\sum_{j=0}^{t} q^{t-j}\beta\mathbb{E}[g(z_j, \theta_j, O_j)]$$

$$+ 2\sum_{j=0}^{t} q^{t-j}\langle z_j, \omega(\theta_j) - \omega(\theta_{j+1})\rangle + \beta^2 C_1 \sum_{j=0}^{t} q^{t-j}, \tag{79}$$

where

$$\begin{aligned} q &\triangleq 1 + 3\beta^2 - 2\beta\lambda, \\ f(z_j, O_j) &\triangleq \langle z_j, (C - \phi_j\phi_j^\top) z_j\rangle, \\ g(z_j, \theta_j, O_j) &\triangleq \langle z_j, \delta_j(\theta_j)\phi_j - \phi_j\phi_j^\top \omega(\theta_j)\rangle. \end{aligned} \tag{80}$$

To simplify notations, let

$$\theta_{t+1} \leftarrow \theta_t + \alpha G_t(\theta_t, \omega_t), \tag{81}$$

$$\omega_{t+1} \leftarrow \omega_t + \beta H_t(\theta_t, \omega_t), \tag{82}$$

where $G_t(\theta, \omega) = \delta_t(\theta)\phi_t - \gamma\left((1-R)\phi_{t+1} + R\frac{\sum_i e^{\varrho\theta^\top\phi_i}\phi_i}{\sum_j e^{\varrho\theta^\top\phi_j}}\right)\phi_t^\top\omega$, and $H_t(\theta, \omega) = (\delta_t(\theta_t) - \phi_t^\top\omega_t)\phi_t$.

We have

$$\|G_t(\theta, \omega)\| \le C_\delta + K\gamma \triangleq C_G. \tag{83}$$

The upper bound of $H_t(\theta, \omega)$ is straightforward:

$$\|H_t(\theta, \omega)\| \le C_\delta + K \triangleq C_H. \tag{84}$$

With these two bounds we can then find the upper bound of the update of tracking error:

$$\begin{aligned} \|z_{t+1} - z_t\| &\le \|H_t(\theta_t, \omega_t)\| + \|\omega(\theta_{t+1}) - \omega(\theta_t)\| \\ &\overset{(a)}{\le} \beta C_H + \alpha\frac{C_\delta}{\lambda}\|G_t(\theta_t, \omega_t)\| \\ &\le \beta C_H + \alpha\frac{C_\delta C_G}{\lambda}, \end{aligned} \tag{85}$$

where (a) is from the Lipschitz of $\omega(\theta)$: $\|\omega(\theta_{t+1}) - \omega(\theta_t)\| \le \frac{L_\delta}{\lambda}\|\theta_{t+1} - \theta_t\| \le \frac{\alpha L_\delta}{\lambda}\|G_t(\theta_t, \omega_t)\|$. Then for the Lipschitz smoothness of function g in (80), it is straightforward to see that

$$\begin{aligned} &|g(\theta, z, O_t) - g(\theta', z', O_t)| \\ &= \langle z, \delta_j(\theta)\phi_j - \phi_j\phi_j^\top\omega(\theta)\rangle - \langle z', \delta_j(\theta')\phi_j - \phi_j\phi_j^\top\omega(\theta')\rangle \\ &= \langle z, \delta_j(\theta)\phi_j - \phi_j\phi_j^\top\omega(\theta)\rangle - \langle z, \delta_j(\theta')\phi_j - \phi_j\phi_j^\top\omega(\theta')\rangle \\ &\quad + \langle z, \delta_j(\theta')\phi_j - \phi_j\phi_j^\top\omega(\theta')\rangle - \langle z', \delta_j(\theta')\phi_j - \phi_j\phi_j^\top\omega(\theta')\rangle \\ &\le K_z L_\delta\left(1 + \frac{1}{\lambda}\right)\|\theta - \theta'\| + C_\delta\left(1 + \frac{1}{\lambda}\right)\|z - z'\|, \end{aligned} \tag{86}$$

where $K_z \triangleq K + \frac{C_\delta}{\lambda}$ being a rough bound on the track error. Also it can be shown that

$$\begin{aligned} |f(z, O_t) - f(z', O_t)| &= \langle z, (C - \phi_t\phi_t^\top)z\rangle - \langle z', (C - \phi_t\phi_t^\top)z'\rangle \\ &= \langle z, (C - \phi_t\phi_t^\top)z\rangle - \langle z, (C - \phi_t\phi_t^\top)z'\rangle \\ &\quad + \langle z, (C - \phi_t\phi_t^\top)z'\rangle - \langle z', (C - \phi_t\phi_t^\top)z'\rangle \\ &\le 4K_z\|z - z'\|. \end{aligned} \tag{87}$$

It is easy to see that

$$\|G_i(\theta, \omega_1) - G_i(\theta, \omega_2)\| \le (\gamma + 2\gamma R\varrho K)\|\omega_1 - \omega_2\|. \tag{88}$$

With these bounds and Lipschitz constants, the following two lemmas can be proved using the similar method of decoupling the Markovian noise in [Wang and Zou, 2020, Bhandari et al., 2018, Zou et al., 2019].

Lemma 7. *Define* $\tau_\beta = \min\{k : m\rho^k \le \beta\}$. *If* $t < \tau_\beta$, *then*

$$\mathbb{E}[f(z_t, O_t)] \le 4K_z^2; \tag{89}$$

and if $t \ge \tau_\beta$, *then*

$$\mathbb{E}[f(z_t, O_t)] \le m_f\beta + m'_f\tau_\beta\beta, \tag{90}$$

where $m_f = 8K_z^2$ *and* $m'_f = 8K_z\left(C_H + \frac{C_G C_\delta}{\lambda}\right)$.

A similar result on $\mathbb{E}[g(\theta_t, z_t, O_t)]$ can also be implied:

Lemma 8. *If* $t < \tau_\beta$, *then*

$$\mathbb{E}[g(\theta_t, z_t, O_t)] \le 2K_z\left(1 + \frac{1}{\lambda}\right)C_\delta; \tag{91}$$

and if $t \ge \tau_\beta$, *then*

$$\mathbb{E}[g(\theta_t, z_t, O_t)] \le m_g\beta + m'_g\tau_\beta\beta, \tag{92}$$

where $m_g = 4K_z\left(1 + \frac{1}{\lambda}\right)C_\delta$ *and* $m'_g = 4K_z L_\delta C_G\left(1 + \frac{1}{\lambda}\right) + C_\delta\left(1 + \frac{1}{\lambda}\right)\left(C_H + \frac{C_G C_\delta}{\lambda}\right)$.

One more lemma is needed to bound the tracking error.

Lemma 9. *Define* $h(\theta, z, O_t) = \left\langle z, -\nabla\omega(\theta)\left(G_t(\theta, \omega(\theta)) + \frac{\nabla J(\theta)}{2}\right)\right\rangle$, *then if* $t < \tau_\beta$,

$$\mathbb{E}[h(\theta_t, z_t, O_t)] \le K_z C_h; \tag{93}$$

and if $t \ge \tau_\beta$,

$$\mathbb{E}[h(\theta_t, z_t, O_t)] \le m_h\beta + m'_h\tau_\beta\beta, \tag{94}$$

where $m_h = 2K_z C_h$ *and* $m'_h = C_h\left(C_H + \frac{C_\delta C_G}{\lambda}\right) + K_z L_h C_G$.

Proof. First we show the Lipschitz smoothness of h as follows. For any θ, θ', z and z', we have that

$$\begin{aligned}
&h(\theta, z, O_t) - h(\theta', z', O_t)\\
&= \left\langle z, -\nabla\omega(\theta)\left(G_t(\theta, \omega(\theta)) + \frac{\nabla J(\theta)}{2}\right)\right\rangle - \left\langle z', -\nabla\omega(\theta')\left(G_t(\theta', \omega(\theta')) + \frac{\nabla J(\theta')}{2}\right)\right\rangle\\
&= \left\langle z, -\nabla\omega(\theta)\left(G_t(\theta, \omega(\theta)) + \frac{\nabla J(\theta)}{2}\right)\right\rangle - \left\langle z', -\nabla\omega(\theta)\left(G_t(\theta, \omega(\theta)) + \frac{\nabla J(\theta)}{2}\right)\right\rangle\\
&\quad + \left\langle z', -\nabla\omega(\theta)\left(G_t(\theta, \omega(\theta)) + \frac{\nabla J(\theta)}{2}\right)\right\rangle - \left\langle z', -\nabla\omega(\theta')\left(G_t(\theta', \omega(\theta')) + \frac{\nabla J(\theta')}{2}\right)\right\rangle.
\end{aligned} \tag{95}$$

We note that

$$\begin{aligned}
&\left\|-\nabla\omega(\theta)\left(G_t(\theta, \omega(\theta)) + \frac{\nabla J(\theta)}{2}\right)\right\|\\
&\le \frac{L_\delta}{\lambda}\left(C_\delta + \gamma(1 - R) + 2\varrho K\gamma R\frac{C_\delta}{\lambda} + \frac{2L_\delta C_\delta}{\lambda}\right) \triangleq C_h,
\end{aligned} \tag{96}$$

and

$$\begin{aligned}
&\left\|-\nabla\omega(\theta)\left(G_t(\theta, \omega(\theta)) + \frac{\nabla J(\theta)}{2}\right) + \nabla\omega(\theta')\left(G_t(\theta', \omega(\theta')) + \frac{\nabla J(\theta')}{2}\right)\right\|\\
&\le \left(\frac{L'_\delta}{L_\delta}C_h + \frac{L_\delta L_{G^*}}{\lambda} + \frac{L_\delta L_J}{2\lambda}\right)\|\theta - \theta'\| \triangleq L_h\|\theta - \theta'\|.
\end{aligned} \tag{97}$$

Hence we have that

$$h(\theta, z, O_t) - h(\theta', z', O_t) \le C_h\|z - z'\| + K_z L_h\|\theta - \theta'\|. \tag{98}$$

We have shown before in (85) that

$$\|z_{t+1} - z_t\| \leq \beta C_H + \alpha \frac{C_\delta C_G}{\lambda}. \tag{99}$$

Hence, we have that

$$|h(\theta_t, z_t, O_t) - h(\theta_{t-\tau}, z_{t-\tau}, O_t)| \leq C_h \left(\beta C_H + \alpha \frac{C_\delta C_G}{\lambda}\right) \tau + K_z L_h C_G \tau \alpha. \tag{100}$$

Define an independent random variable $\hat{O} = (\hat{S}, \hat{A}, \hat{S}') \sim \mu_\pi \times \mathsf{P}(\cdot|\hat{S}, \hat{A})$, then we have

$$\mathbb{E}_{\hat{O}}[h(\theta, z, \hat{O})] = 0 \tag{101}$$

for any θ and z. Thus by uniform ergodicity, we have that

$$\mathbb{E}[h(\theta_{t-\tau}, z_{t-\tau}, O_t)] \leq \mathbb{E}[h(\theta_{t-\tau}, z_{t-\tau}, O_t)] - \mathbb{E}_{\hat{O}}[h(\theta_t, z_t, \hat{O})] \leq 2K_z C_h m \rho^\tau. \tag{102}$$

Then if $t \leq \tau_\beta$, we have the straightforward bound

$$\mathbb{E}[h(\theta_t, z_t, O_t)] \leq K_z C_h; \tag{103}$$

and if $t > \tau_\beta$, we have that

$$\begin{aligned}
\mathbb{E}[h(\theta_t, z_t, O_t)] &\leq \mathbb{E}[h(\theta_{t-\tau_\beta}, z_{t-\tau_\beta}, O_t)] + C_h \left(\beta C_H + \alpha \frac{C_\delta C_G}{\lambda}\right) \tau_\beta + K_z L_h C_G \tau_\beta \alpha \\
&\leq 2K_z C_h m \rho^{\tau_\beta} + C_h \left(\beta C_H + \alpha \frac{C_\delta C_G}{\lambda}\right) \tau_\beta + K_z L_h C_G \tau_\beta \alpha \\
&\triangleq m_h \beta + m'_h \tau_\beta \beta,
\end{aligned} \tag{104}$$

where $m_h = 2K_z C_h$ and $m'_h = C_h \left(C_H + \frac{C_\delta C_G}{\lambda}\right) + K_z L_h C_G$. This completes the proof. □

Now we bound the tracking error in (79). We first rewrite it as

$$\begin{aligned}
\mathbb{E}[\|z_{t+1}\|^2] \leq q^{t+1}\|z_0\|^2 &+ \underbrace{2\sum_{j=0}^{t} q^{t-j}\beta \mathbb{E}[f(z_j, O_j)]}_{A_t} + \underbrace{2\sum_{j=0}^{t} q^{t-j}\beta \mathbb{E}[g(z_j, \theta_j, O_j)]}_{B_t} \\
&+ \underbrace{2\sum_{j=0}^{t} q^{t-j}\langle z_j, \omega(\theta_j) - \omega(\theta_{j+1})\rangle}_{C_t} + \beta^2 C_1 \sum_{j=0}^{t} q^{t-j}.
\end{aligned} \tag{105}$$

The second term A_t can be bounded as follows:

$$\begin{aligned}
A_t &= 2\sum_{j=0}^{t} q^{t-j}\beta \mathbb{E}[f(z_j, O_j)] \\
&= 2\sum_{j=0}^{\tau_\beta - 1} q^{t-j}\beta \mathbb{E}[f(z_j, O_j)] + 2\sum_{j=\tau_\beta}^{t} q^{t-j}\beta \mathbb{E}[f(z_j, O_j)] \\
&\leq 8\sum_{j=0}^{\tau_\beta - 1} q^{t-j} K_z \beta + 2\sum_{j=\tau_\beta}^{t} q^{t-j}\beta(m_f \beta + m'_f \tau_\beta \beta) \\
&\leq 16 K_z \beta \frac{q^{t+1-\tau_\beta}}{1-q} + 2\beta(m_f \beta + m'_f \tau_\beta \beta)\frac{1 - q^{t-\tau_\beta+1}}{1-q}.
\end{aligned} \tag{106}$$

Similarly, we have that

$$B_t \leq 4K_z \beta \left(1 + \frac{1}{\lambda}\right) C_\delta \frac{q^{t+1-\tau_\beta}}{1-q} + 2\beta(m_g \beta + m'_g \tau_\beta \beta)\frac{1 - q^{t-\tau_\beta+1}}{1-q}. \tag{107}$$

For C_t, we first note that

$$
\begin{aligned}
&\mathbb{E}\left[\langle z_i, \omega\left(\theta_i\right)-\omega\left(\theta_{i+1}\right)\rangle\right] \\
&\overset{(a)}{=} \mathbb{E}\left[\langle z_i, \nabla\omega\left(\theta_i\right)\left(\theta_i-\theta_{i+1}\right)+R_2\rangle\right] \\
&= \mathbb{E}\left[\langle z_i, -\alpha\nabla\omega\left(\theta_i\right) G_i\left(\theta_i,\omega_i\right)+R_2\rangle\right] \\
&= \mathbb{E}\Bigg[\Bigg\langle z_i, -\alpha\nabla\omega\left(\theta_i\right)\left(G_i\left(\theta_i,\omega_i\right)-G_i\left(\theta_i,\omega\left(\theta_i\right)\right)+G_i\left(\theta_i,\omega\left(\theta_i\right)\right)+\frac{\nabla J\left(\theta_i\right)}{2}-\frac{\nabla J\left(\theta_i\right)}{2}\right) \\
&\quad +R_2\Bigg\rangle\Bigg] \\
&= \underbrace{\mathbb{E}\left[\left\langle z_i, -\alpha\nabla\omega\left(\theta_i\right)\left(G_i\left(\theta_i,\omega\left(\theta_i\right)\right)+\frac{\nabla J\left(\theta_i\right)}{2}\right)\right\rangle\right]}_{(b)} \\
&\quad + \underbrace{\mathbb{E}\left[\left\langle z_i, -\alpha\nabla\omega\left(\theta_i\right)\left(G_i\left(\theta_i,\omega_i\right)-G_i\left(\theta_i,\omega\left(\theta_i\right)\right)-\frac{\nabla J\left(\theta_i\right)}{2}\right)+R_2\right\rangle\right]}_{(c)}, \qquad (108)
\end{aligned}
$$

where (a) follows from the Taylor expansion, and R_2 is the remaining term with norm $\|R_2\| = \mathcal{O}(\alpha^2)$. Term (b) can be bounded using Lemma 9, where

$$
\mathbb{E}\left[\left\langle z_i, -\alpha\nabla\omega\left(\theta_i\right)\left(G_i\left(\theta_i,\omega\left(\theta_i\right)\right)+\frac{\nabla J\left(\theta_i\right)}{2}\right)\right\rangle\right] = \alpha\mathbb{E}\left[h\left(\theta_i, z_i, O_i\right)\right]. \qquad (109)
$$

Term (c) can be bounded as follows.

$$
\begin{aligned}
&\left\langle z_i, -\alpha\nabla\omega\left(\theta_i\right)\left(G_i\left(\theta_i,\omega_i\right)-G_i\left(\theta_i,\omega\left(\theta_i\right)\right)-\frac{\nabla J\left(\theta_i\right)}{2}\right)+R_2\right\rangle \\
&\overset{(d)}{\leq} \frac{\lambda\beta}{8}\|z_i\|^2+\frac{2}{\lambda\beta}\left\|\alpha\nabla\omega\left(\theta_i\right)\left(G_i\left(\theta_i,\omega_i\right)-G_i\left(\theta_i,\omega\left(\theta_i\right)\right)-\frac{\nabla J\left(\theta_i\right)}{2}\right)+R_2\right\|^2 \\
&\leq \frac{\lambda\beta}{8}\|z_i\|^2 \\
&\quad +\frac{6}{\lambda\beta}\left(\left\|\alpha\nabla\omega\left(\theta_i\right)\left(G_i\left(\theta_i,\omega_i\right)-G_i\left(\theta_i,\omega\left(\theta_i\right)\right)\right)\right\|^2+\left\|\alpha\nabla\omega\left(\theta_i\right)\frac{\nabla J\left(\theta_i\right)}{2}\right\|^2+\|R_2\|^2\right) \\
&\leq \frac{\lambda\beta}{8}\|z_i\|^2+\frac{6\alpha^2}{\lambda\beta}\frac{L_\delta^2}{\lambda^2}(\gamma+2\gamma R\varrho K)^2\|z_i\|^2+\frac{3\alpha^2}{2\lambda\beta}\frac{L_\delta^2}{\lambda^2}\|\nabla J(\theta_i)\|^2+\frac{6}{\lambda\beta}\|R_2\|^2. \qquad (110)
\end{aligned}
$$

where (d) is from $\langle x, y\rangle \leq \frac{\lambda\beta}{8}\|x\|^2+\frac{2}{\lambda\beta}\|y\|^2$ for any $x, y \in \mathbb{R}^N$ and the fact that $\|G_i(\theta,\omega_1)-G_i(\theta,\omega_2)\| \leq (\gamma+2\gamma\varrho RK)\|\omega_1-\omega_2\|$ for any $\|\theta\| \leq R$ and ω_1, ω_2, which is from (88) .

Finally the term C_t can be bounded as follows.

$$
\begin{aligned}
C_t &= 2\sum_{j=0}^{t} q^{t-j}\langle z_j, \omega(\theta_j)-\omega(\theta_{j+1})\rangle \\
&= 2\sum_{j=0}^{t} q^{t-j}\alpha\mathbb{E}[h(\theta_j, z_j, O_j)] \\
&\quad +2\sum_{j=0}^{t} q^{t-j}\left(\frac{\lambda\beta}{8}\|z_i\|^2+\frac{6\alpha^2}{\lambda\beta}\frac{L_\delta^2}{\lambda^2}(\gamma+2\gamma R\varrho K)^2\|z_i\|^2+\frac{3\alpha^2}{2\lambda\beta}\frac{L_\delta^2}{\lambda^2}\|\nabla J(\theta_i)\|^2+\frac{6}{\lambda\beta}\|R_2\|^2\right) \\
&\triangleq 2\sum_{j=0}^{t} q^{t-j}\alpha\mathbb{E}[h(\theta_j, z_j, O_j)]+M_t, \qquad (111)
\end{aligned}
$$

where $M_t = 2\sum_{j=0}^{t} q^{t-j}\left(\frac{\lambda\beta}{8}\|z_i\|^2 + \frac{6\alpha^2}{\lambda\beta}\frac{L_\delta^2}{\lambda^2}(\gamma + 2\gamma R\varrho K)^2\|z_i\|^2 + \frac{3\alpha^2}{2\lambda\beta}\frac{L_\delta^2}{\lambda^2}\|\nabla J(\theta_i)\|^2 + \frac{6}{\lambda\beta}\|R_2\|^2\right)$. From Lemma 9, we have that

$$
\begin{aligned}
&2\sum_{j=0}^{t} q^{t-j}\alpha\mathbb{E}[h(\theta_j, z_j, O_j)] \\
&\leq 2\alpha\left(\sum_{j=0}^{\tau_\beta-1} q^{t-j}\mathbb{E}[h(\theta_j, z_j, O_j)] + \sum_{j=\tau_\beta}^{t} q^{t-j}\mathbb{E}[h(\theta_j, z_j, O_j)]\right) \\
&\leq 4K_z C_h\alpha\sum_{j=0}^{\tau_\beta-1} q^{t-j} + 2\alpha(m_h\beta + m_h'\tau_\beta\beta)\sum_{j=\tau_\beta}^{t} q^{t-j} \\
&= 4K_z C_h\alpha\frac{q^{t+1-\tau_\beta}}{1-q} + 2\alpha(m_h\beta + m_h'\tau_\beta\beta)\frac{1-q^{t-\tau_\beta+1}}{1-q}, \qquad (112)
\end{aligned}
$$

and this implies that

$$
C_t \leq 4K_z C_h\alpha\frac{q^{t+1-\tau_\beta}}{1-q} + 2\alpha(m_h\beta + m_h'\tau_\beta\beta)\frac{1-q^{t-\tau_\beta+1}}{1-q} + M_t. \qquad (113)
$$

Now we plug the bounds on A_t, B_t and C_t in (79), we have that

$$
\begin{aligned}
&\mathbb{E}[\|z_{t+1}\|^2] \\
&\leq q^{t+1}\|z_0\|^2 + \beta^2 C_1\frac{1-q^{t+1}}{1-q} + \left(16K_z\beta + 4K_z C_\delta\beta\left(1+\frac{1}{\lambda}\right) + 4K_z C_h\alpha\right)\frac{q^{t+1-\tau_\beta}}{1-q} \\
&\quad + \left(2\beta(m_f\beta + m_f'\tau_\beta\beta) + 2\beta(m_g\beta + m_g'\tau_\beta\beta) + 2\alpha(m_h\beta + m_h'\tau_\beta\beta)\right)\frac{1-q^{t-\tau_\beta+1}}{1-q} + M_t \\
&\leq q^{t+1}\|z_0\|^2 + \beta^2 C_1\frac{1-q^{t+1}}{1-q} + C_z\beta\frac{q^{t+1-\tau_\beta}}{1-q} + \beta(m_z\beta + m_z'\tau_\beta\beta)\frac{1-q^{t-\tau_\beta+1}}{1-q} + M_t, \qquad (114)
\end{aligned}
$$

where $C_z = 16K_z + 4K_z C_\delta\left(1+\frac{1}{\lambda}\right) + 4K_z C_h\frac{\alpha}{\beta}$, $m_z = 2m_f + 2m_g + 2\frac{\alpha}{\beta}m_h$ and $m_z' = 2m_f' + 2m_g' + \frac{2\alpha}{\beta}m_h'$. Note that $q = 1 + 3\beta^2 - 2\beta\lambda \triangleq 1 - u\beta \leq e^{-u\beta}$, where $u = 2\lambda - 3\beta$. Hence it implies that

$$
\begin{aligned}
&\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|z_t\|^2]}{T} \\
&\leq \frac{1}{T}\left(\frac{\|z_0\|^2}{1-e^{-u\beta}} + \beta^2 C_1\frac{T}{u\beta} + 4K_z^2\tau_\beta\right. \\
&\quad \left. + \sum_{t=\tau_\beta-1}^{T-1}\left(C_z\beta\frac{q^{t+1-\tau_\beta}}{u\beta} + \beta(m_z\beta + m_z'\tau_\beta\beta)\frac{1-q^{t-\tau_\beta+1}}{u\beta} + M_t\right)\right) \\
&\leq \frac{1}{T}\left(\frac{\|z_0\|^2}{1-e^{-u\beta}} + \beta^2 C_1\frac{T}{u\beta} + 4K_z^2\tau_\beta\right. \\
&\quad \left. + c_z\beta\frac{\sum_{t=0}^{T-1} e^{-ut\beta}}{u\beta} + \beta(m_z\beta + m_z'\tau_\beta\beta)\frac{T}{u\beta} + \sum_{t=0}^{T-1} M_t\right) \\
&\leq \frac{1}{T}\left(\frac{\|z_0\|^2}{1-e^{-u\beta}} + \beta^2 C_1\frac{T}{u\beta} + 4K_z^2\tau_\beta + c_z\beta\frac{1}{(u\beta)(1-e^{-u\beta})} + \beta(m_z\beta + m_z'\tau_\beta\beta)\frac{T}{u\beta}\right. \\
&\quad \left. + \sum_{t=0}^{T-1} M_t\right) \\
&= \frac{1}{T}\left(\frac{\|z_0\|^2}{1-e^{-u\beta}} + \beta C_1\frac{T}{u} + 4K_z^2\tau_\beta + \frac{c_z}{u(1-e^{-u\beta})} + (m_z\beta + m_z'\tau_\beta\beta)\frac{T}{u} + \sum_{t=0}^{T-1} M_t\right)
\end{aligned}
$$

$$
\begin{aligned}
&\leq \frac{\|z_0\|^2}{T(1-e^{-u\beta})} + \beta\frac{C_1}{u} + 4K_z^2\frac{\tau_\beta}{T} + \frac{c_z}{u(1-e^{-u\beta})T} + (m_z\beta + m_z'\tau_\beta\beta)\frac{1}{u} + \frac{\sum_{t=0}^{T-1} M_t}{T} \\
&\triangleq Q_T + \frac{\sum_{t=0}^{T-1} M_t}{T} \\
&= \mathcal{O}\left(\frac{1}{T\beta} + \beta\tau_\beta + \frac{\tau_\beta}{T} + \frac{\sum_{t=0}^{T-1} M_t}{T}\right), \qquad (115)
\end{aligned}
$$

where $Q_T = \frac{\|z_0\|^2}{T(1-e^{-u\beta})} + \beta\frac{C_1}{u} + 4K_z^2\frac{\tau_\beta}{T} + \frac{c_z}{u(1-e^{-u\beta})T} + (m_z\beta + m_z'\tau_\beta\beta)\frac{1}{u}$.

We then compute $\sum_{t=0}^{T-1} M_t$. Recall that $M_t = 2\sum_{j=0}^{t} q^{t-j}\Big(\frac{\lambda\beta}{8}\|z_i\|^2 + \frac{6\alpha^2}{\lambda\beta}\frac{L_\delta^2}{\lambda^2}(\gamma + 2\gamma R\varrho K)^2\|z_i\|^2 + \frac{3\alpha^2}{2\lambda\beta}\frac{L_\delta^2}{\lambda^2}\|\nabla J(\theta_i)\|^2 + \frac{6}{\lambda\beta}\|R_2\|^2\Big)$. From double sum trick, i.e., $\sum_{t=0}^{T-1}\sum_{i=0}^{t} e^{-u(t-i)\beta}x_i \leq \frac{1}{1-e^{-u\beta}}\sum_{t=0}^{T-1} x_t$ for any $x_t \geq 0$, we have that

$$
\begin{aligned}
\sum_{t=0}^{T-1} M_t \leq{}& \frac{2}{1-e^{-u\beta}}\left(\frac{\lambda\beta}{8} + \frac{6\alpha^2}{\lambda\beta}\frac{L_\delta^2}{\lambda^2}(\gamma + 2\gamma R\varrho K)^2\right)\sum_{t=0}^{T-1}\mathbb{E}[\|z_t\|^2] \\
&+ \frac{2}{1-e^{-u\beta}}\frac{3\alpha^2}{2\lambda\beta}\frac{L_\delta^2}{\lambda^2}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2] + \frac{6}{\lambda\beta}\frac{2}{1-e^{-u\beta}}\|R_2\|^2 T. \qquad (116)
\end{aligned}
$$

Note that $1 - e^{-u\beta} = \mathcal{O}(\beta)$, thus we can choose α and β such that $\frac{2}{1-e^{-u\beta}}\left(\frac{\lambda\beta}{8} + \frac{6\alpha^2}{\lambda\beta}\frac{L_\delta^2}{\lambda^2}(\gamma + 2\gamma R\varrho K)^2\right) \leq \frac{1}{2}$, then by plugging $\sum_{t=0}^{T-1} M_t$ in (115) we have that

$$
\frac{1}{2}\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|z_t\|^2]}{T} \leq Q_T + \frac{2}{1-e^{-u\beta}}\frac{3\alpha^2}{2\lambda\beta}\frac{L_\delta^2}{\lambda^2}\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}{T} + \frac{6}{\lambda\beta}\frac{2}{1-e^{-u\beta}}\|R_2\|^2, \qquad (117)
$$

and this implies that

$$
\begin{aligned}
\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|z_t\|^2]}{T} &\leq 2Q_T + \frac{2}{1-e^{-u\beta}}\frac{3\alpha^2}{\lambda\beta}\frac{L_\delta^2}{\lambda^2}\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}{T} + \frac{6}{\lambda\beta}\frac{4}{1-e^{-u\beta}}\|R_2\|^2 \\
&= \mathcal{O}\left(\frac{1}{T\beta} + \beta\tau_\beta + \frac{\alpha^2}{\beta^2}\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}{T}\right), \qquad (118)
\end{aligned}
$$

which completes the development of error bound on the tracking error.

D.3 Finite-Time Error Bound

Now with the tracking error in (118), we derive the finite-time error of the robust TDC. From Lemma 6 and Taylor expansion, we have that

$$
\begin{aligned}
J(\theta_{t+1}) &\leq J(\theta_t) + \langle\nabla J(\theta_t), \theta_{t+1} - \theta_t\rangle + \frac{L_J}{2}\|\theta_{t+1} - \theta_t\|^2 \\
&= J(\theta_t) + \alpha\langle\nabla J(\theta_t), G_t(\theta_t, \omega_t)\rangle + \frac{L_J}{2}\alpha^2||G_t(\theta_t, \omega_t)||^2 \\
&= J(\theta_t) - \alpha\left\langle\nabla J(\theta_t), -G_t(\theta_t, \omega_t) - \frac{\nabla J(\theta_t)}{2} + G_t(\theta_t, \omega(\theta_t)) - G_t(\theta_t, \omega(\theta_t))\right\rangle \\
&\quad - \frac{\alpha}{2}||\nabla J(\theta_t)||^2 + \frac{L_J}{2}\alpha^2||G_t(\theta_t, \omega_t)||^2 \\
&= J(\theta_t) - \alpha\langle\nabla J(\theta_t), -G_t(\theta_t, \omega_t) + G_t(\theta_t, \omega(\theta_t))\rangle \\
&\quad + \alpha\left\langle\nabla J(\theta_t), \frac{\nabla J(\theta_t)}{2} + G_t(\theta_t, \omega(\theta_t))\right\rangle - \frac{\alpha}{2}||\nabla J(\theta_t)||^2 + \frac{L_J}{2}\alpha^2||G_t(\theta_t, \omega_t)||^2 \\
&\leq J(\theta_t) + \alpha\|\nabla J(\theta_t)\|(\gamma + 2\gamma RK\varrho)\|\omega(\theta_t) - \omega_t\| - \frac{\alpha}{2}||\nabla J(\theta_t)||^2
\end{aligned}
$$

$$+\alpha\left\langle\nabla J(\theta_t),\frac{\nabla J(\theta_t)}{2}+G_t(\theta_t,\omega(\theta_t))\right\rangle+\frac{L_J}{2}\alpha^2||G_t(\theta_t,\omega_t)||^2. \tag{119}$$

By taking expectation on both sides and summing up from 0 to $T-1$, we have that

$$\begin{aligned}
&\sum_{t=0}^{T-1}\frac{\alpha}{2}\mathbb{E}[\|\nabla J(\theta_t)\|^2]\\
&\leq J(\theta_0)-J(\theta_T)+\alpha(\gamma+2\gamma RK\varrho)\sqrt{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}\sqrt{\sum_{t=0}^{T-1}\mathbb{E}[\|z_t\|^2]}\\
&\quad+\sum_{t=0}^{T-1}\alpha\mathbb{E}\left[\left\langle\nabla J(\theta_t),\frac{\nabla J(\theta_t)}{2}+G_t(\theta_t,\omega(\theta_t))\right\rangle\right]+\frac{L_J}{2}\sum_{t=0}^{T-1}\alpha^2\mathbb{E}[\|G_t(\theta_t,\omega_t)\|^2],
\end{aligned} \tag{120}$$

which follows from the Cauchy-Schwartz inequality: $\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|\|z_t\|] \leq \sum_{t=0}^{T-1}\sqrt{\mathbb{E}[\|\nabla J(\theta_t)\|^2]\mathbb{E}[\|z_t\|^2]} \leq \sqrt{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}\sqrt{\sum_{t=0}^{T-1}\mathbb{E}[\|z_t\|^2]}$. To bound the Markovian noise term, i.e., $\left\langle\nabla J(\theta),\frac{\nabla J(\theta)}{2}+G_t(\theta,\omega(\theta))\right\rangle$, we first need some bounds and smoothness conditions. It can be shown that

$$\|G_t(\theta,\omega(\theta))\|\leq C_\delta+\frac{C_\delta}{\lambda}(\gamma+2\varrho K\gamma R)\triangleq C_{G*}, \tag{121}$$

$$\|G_t(\theta,\omega(\theta))-G_t(\theta',\omega(\theta'))\|\leq\left(L_\delta+\frac{L_\delta}{\lambda}(\gamma+2\gamma R\varrho K)+\frac{C_\delta}{\lambda}L'_\delta\right)\|\theta-\theta'\|\triangleq L_{G*}\|\theta-\theta'\|. \tag{122}$$

Lemma 10. *Define* $\zeta(\theta,O_t)\triangleq\left\langle\nabla J(\theta),\frac{\nabla J(\theta)}{2}+G_t(\theta,\omega(\theta))\right\rangle$, *and let* $\tau_\alpha\triangleq\min\left\{k:m\rho^k\leq\alpha\right\}$. *If* $t<\tau_\alpha$, *then*

$$\mathbb{E}[\zeta(\theta_t,O_t)]\leq\frac{C_\delta L_\delta}{\lambda}\left(\frac{C_\delta L_\delta}{2\lambda}+C_{G*}\right)\triangleq C_\zeta; \tag{123}$$

and if $t\geq\tau_\alpha$, *then*

$$\mathbb{E}[\zeta(\theta_t,O_t)]\leq m_\zeta\alpha+m'_\zeta\tau_\alpha\alpha, \tag{124}$$

where $m_\zeta=2C_\zeta$ *and* $m'_\zeta=C_G\left(\frac{L_JC_\delta L_\delta}{\lambda}+\frac{C_\delta L_\delta L_{G*}}{\lambda}+L_JC_{G*}\right)$.

Next we plug the tracking error (118) in (120).

$$\begin{aligned}
&\sum_{t=0}^{T-1}\frac{\alpha}{2}\mathbb{E}[\|\nabla J(\theta_t)\|^2]\\
&\leq J(\theta_0)-J(\theta_T)+\alpha(\gamma+2\gamma RK\varrho)\sqrt{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}\sqrt{2TQ_T+2\sum_{t=0}^{T-1}M_t}\\
&\quad+\alpha\tau_\alpha C_\zeta+\alpha^2(T-\tau_\alpha)(m_\zeta+m'_\zeta\tau_\alpha)+\frac{L_J}{2}\alpha^2C_G^2T.
\end{aligned} \tag{125}$$

Divided both sides by $\frac{\alpha T}{2}$, we have that

$$\begin{aligned}
&\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}{T}\\
&\leq\frac{2J(\theta_0)-2J(\theta_T)}{\alpha T}+2(\gamma+2\gamma RK\varrho)\sqrt{\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}{T}}\sqrt{2Q_T+2\frac{\sum_{t=0}^{T-1}M_t}{T}}\\
&\quad+\frac{2\tau_\alpha C_\zeta}{T}+2\alpha(m_\zeta+m'_\zeta\tau_\alpha)+L_J\alpha C_G^2.
\end{aligned} \tag{126}$$

We know from (118) that $2\frac{\sum_{t=0}^{T-1} M_t}{T} \leq \frac{2}{1-e^{-u\beta}}\frac{3\alpha^2}{\lambda\beta}\frac{L_\delta^2}{\lambda^2}\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}{T} + \frac{6}{\lambda\beta}\frac{4}{1-e^{-u\beta}}\|R_2\|^2$, thus

$$
\begin{aligned}
&\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}{T} \\
&\leq \frac{2J(\theta_0)-2J(\theta_T)}{\alpha T} + 2(\gamma+2\gamma RK\varrho)\sqrt{\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}{T}} \\
&\quad\left(\sqrt{2Q_T + \frac{6}{\lambda\beta}\frac{4}{1-e^{-u\beta}}\|R_2\|^2} + \sqrt{\frac{2}{1-e^{-u\beta}}\frac{3\alpha^2}{\lambda\beta}\frac{L_\delta^2}{\lambda^2}\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}{T}}\right) \\
&\quad + \frac{2\tau_\alpha C_\zeta}{T} + 2\alpha(m_\zeta + m'_\zeta\tau_\alpha) + L_J\alpha C_G^2 \\
&= \frac{2J(\theta_0)-2J(\theta_T)}{\alpha T} + 2(\gamma+2\gamma RK\varrho)\sqrt{\frac{2}{1-e^{-u\beta}}\frac{3\alpha^2}{\lambda\beta}\frac{L_\delta^2}{\lambda^2}}\sqrt{\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}{T}} \\
&\quad\left(\sqrt{2Q_T + \frac{6}{\lambda\beta}\frac{4}{1-e^{-u\beta}}\|R_2\|^2}\right)2(\gamma+2\gamma RK\varrho)\sqrt{\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}{T}} \\
&\quad + \frac{2\tau_\alpha C_\zeta}{T} + 2\alpha(m_\zeta + m'_\zeta\tau_\alpha) + L_J\alpha C_G^2 \\
&\triangleq \frac{2J(\theta_0)-2J(\theta_T)}{\alpha T} + K_1\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}{T} + K_2\sqrt{\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}{T}} + \frac{2\tau_\alpha C_\zeta}{T} \\
&\quad + 2\alpha(m_\zeta + m'_\zeta\tau_\alpha) + L_J\alpha C_G^2, \qquad (127)
\end{aligned}
$$

where $K_1 = 2(\gamma + 2\gamma RK\varrho)\sqrt{\frac{2}{1-e^{-u\beta}}\frac{3\alpha^2}{\lambda\beta}\frac{L_\delta^2}{\lambda^2}} = \mathcal{O}\left(\frac{\alpha}{\beta}\right)$ and $K_2 = \left(\sqrt{2Q_T + \frac{6}{\lambda\beta}\frac{4}{1-e^{-u\beta}}\|R_2\|^2}\right)2(\gamma + 2\gamma RK\varrho) = \mathcal{O}\left(\sqrt{\frac{\alpha^4}{\beta^2} + \frac{1}{T\beta} + \beta\tau_\beta}\right)$. Thus we can choose α and β such that $K_1 \leq \frac{1}{2}$, then we have that

$$
\begin{aligned}
&\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}{T} \\
&\leq \frac{4J(\theta_0)-4J(\theta_T)}{\alpha T} + 2K_2\sqrt{\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}{T}} + \frac{4\tau_\alpha C_\zeta}{T} + 4\alpha(m_\zeta + m'_\zeta\tau_\alpha) + 2L_J\alpha C_G^2 \\
&\triangleq U + V\sqrt{\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}{T}}, \qquad (128)
\end{aligned}
$$

where $U = \frac{4J(\theta_0)-4J(\theta_T)}{\alpha T} + \frac{4\tau_\alpha C_\zeta}{T} + 4\alpha(m_\zeta + m'_\zeta\tau_\alpha) + 2L_J\alpha C_G^2 = \mathcal{O}(\alpha\tau_\alpha + \frac{1}{\alpha T})$ and $V = 2K_2$. Hence, we have that

$$
\begin{aligned}
&\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}{T} \\
&\leq \left(\frac{V+\sqrt{V^2+4U}}{2}\right)^2 \\
&\overset{(a)}{\leq} V^2 + 2U \\
&\leq 16\left(2Q_T + \frac{6}{\lambda\beta}\frac{4}{1-e^{-u\beta}}\|R_2\|^2\right)(\gamma+2\gamma RK\varrho)^2 + \frac{8J(\theta_0)-8J(\theta_T)}{\alpha T} + \frac{8\tau_\alpha C_\zeta}{T} \\
&\quad + 8\alpha(m_\zeta + m'_\zeta\tau_\alpha) + 4L_J\alpha C_G^2 \\
&= \mathcal{O}\left(\frac{1}{T\alpha} + \alpha\tau_\alpha + \frac{1}{T\beta} + \beta\tau_\beta\right), \qquad (129)
\end{aligned}
$$

where $Q_T = \frac{\|z_0\|^2}{T(1-e^{-u\beta})} + \beta\frac{C_1}{u} + 4K^2\frac{\tau_\beta}{T} + \frac{c_z}{u(1-e^{-u\beta})T} + (m_z\beta + m'_z\tau_\beta\beta)\frac{1}{u}$.

D.4 Constants

In this section we list all the constants occurred in our proof for the readers' reference.

$$C_\delta = c_{\max} + \gamma R \frac{\log|\mathcal{S}|}{\varrho} + (1+\gamma)K, \tag{130}$$

$$L_\delta = (1+\gamma), \tag{131}$$

$$L'_\delta = 2\gamma R\varrho, \tag{132}$$

$$L_J = 2\left(\frac{L_\delta^2}{\lambda} + \frac{C_\delta L'_\delta}{\lambda}\right), \tag{133}$$

$$C_1 = 3\left(C_\delta + \frac{C_\delta}{\lambda}\right)^2 + 3\left(C_\delta + (1+2R\varrho K)\frac{C_\delta}{\lambda}\right)^2, \tag{134}$$

$$C_G = C_\delta + \gamma K + 2\gamma\varrho R K^2, \tag{135}$$

$$C_H = C_\delta + K, \tag{136}$$

$$K_z = K + \frac{C_\delta}{\lambda}, \tag{137}$$

$$m_g = 4K_z\left(1+\frac{1}{\lambda}\right)C_\delta, \tag{138}$$

$$m'_g = 4K_z L_\delta C_G\left(1+\frac{1}{\lambda}\right) + C_\delta\left(1+\frac{1}{\lambda}\right)\left(C_H + \frac{C_G C_\delta}{\lambda}\right), \tag{139}$$

$$m_f = 8K_z^2, \tag{140}$$

$$m'_f = 8K_z\left(C_H + \frac{C_G C_\delta}{\lambda}\right), \tag{141}$$

$$m_h = 2K_z C_h, \tag{142}$$

$$m'_h = C_h\left(C_H + \frac{C_\delta C_G}{\lambda}\right) + K_z L_h C_G, \tag{143}$$

$$C_{G*} = C_\delta + \frac{C_\delta}{\lambda}(\gamma + 2\varrho K\gamma R), \tag{144}$$

$$L_{G*} = L_\delta + \frac{L_\delta}{\lambda}(\gamma + 2\gamma R\varrho K) + \frac{C_\delta}{\lambda}L'_\delta, \tag{145}$$

$$L_h = \frac{L'_\delta}{L_\delta}C_h + \frac{L_\delta L_{G^*}}{\lambda} + \frac{L_\delta L_J}{2\lambda}, \tag{146}$$

$$C_h = \frac{L_\delta}{\lambda}\left(C_\delta + \gamma(1-R) + 2\varrho K\gamma R\frac{C_\delta}{\lambda} + \frac{2L_\delta C_\delta}{\lambda}\right), \tag{147}$$

$$C_\zeta = \frac{C_\delta L_\delta}{\lambda}\left(\frac{C_\delta L_\delta}{2\lambda} + C_{G*}\right), \tag{148}$$

$$m_\zeta = 2C_\zeta, \tag{149}$$

$$m'_\zeta = C_G\left(\frac{L_J C_\delta L_\delta}{\lambda} + \frac{C_\delta L_\delta L_{G*}}{\lambda} + L_J C_{G*}\right) \tag{150}$$

E Experiments

Experiments in Section 6.1:

Frozen Lake Problem. We consider a 4×4 Frozen Lake problem. We set $\gamma = 0.96$, $\alpha = 0.8$.

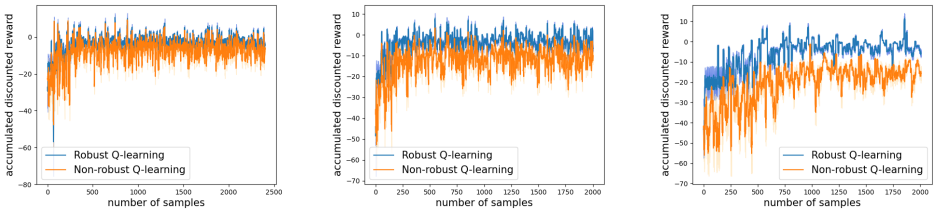
Cart-Pole Problem. We set $\gamma = 0.95$, $\alpha = 0.2$.

Experiments in Section 6.2:

Frozen Lake Problem. We consider a 4×4 Frozen Lake problem. We set $\alpha = 0.1$, $\beta = 0.5$ and $\gamma = 0.9$. The initialization is $\theta = (1, 1, 1, 1, 1) \in \mathbb{R}^5$ and $\omega = (0, 0, 0, 0, 0)$. Each entry of every base function ϕ_s is generated uniformly at random between $(0, 1)$.

Additional Experiments on the Taxi Problem.

We use the same setting as in Section 6.1 to demonstrate the robustness of our robust Q-learning algorithm. For the step size and discount factor, we set $\alpha = 0.3$ and $\gamma = 0.8$. The results are shown in fig. 5, from which the same observation that our robust Q-learning is robust to model uncertainty, and achieves a much higher reward when the mismatch between the training and test MDPs enlarges.



(a) p=0.1, R=0.1 (b) p=0.05, R=0.2 (c) p=0.1, R=0.2

Figure 5: **Taxi-v3**: robust Q-learning v.s. non-robust Q-learning.