Finite Sample Analysis of Average-Reward TD Learning and Q-Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

The focus of this paper is on sample complexity guarantees of average-reward reinforcement learning algorithms, which are known to be more challenging to study than their discounted-reward counterparts. To the best of our knowledge, we provide the first known finite sample guarantees using both constant and diminishing step sizes of (i) average-reward TD(λ) with linear function approximation for policy evaluation and (ii) average-reward Q-learning in the tabular setting to find the optimal policy. A major challenge is that since the value functions are agnostic to an additive constant, the corresponding Bellman operators are no longer contraction mappings under any norm. We obtain the results for TD(λ) by working in an appropriately defined subspace that ensures uniqueness of the solution. For Q-learning, we exploit the span seminorm contractive property of the Bellman operator and construct a novel Lyapunov function obtained by infimal convolution of the generalized Moreau envelop and the indicator function of a set.

1 Introduction

The average-reward setting is a classical setting for formulating the goal in an infinite-horizon Markov decision process (MDP) [1]. The need to maximize the average reward has been demonstrated in many applications, including scheduling automatic guided vehicles [2], inventory management in supply chains [3], communication system control and routing [4] and cooperative multi-robot learning [5]. In these problems, the discounted-reward criterion usually leads to poor long-time performance since the system operates over an extended period of time and the main objective is to perform consistently well over the long run.

Even though there is a well developed theory of average-reward MDPs [6, 7, 8], the theoretical understanding of average-reward RL methods is still quite limited. Most existing results are focused only on asymptotic convergence [9, 10, 11, 12]. The focus of this paper is to understand the sample efficiency. *How much data is required to guarantee a given level of accuracy?*

Recent literature obtains finite sample guarantees for discounted reward TD and Q learning algorithms [13, 14, 15, 16, 17] by developing novel analytical techniques. Such a study of average-reward RL algorithms is not undertaken. Analysis of average-reward RL algorithms is known to be more challenging to study than their discounted-reward counterparts. The key property that is exploited in the study of discounted-reward problems is the contraction property of the underlying Bellman operator. In the average-reward setting, such a contraction property does not hold under any norm, and the solution of the Bellman equation is known to have multiple fixed points.

In this work, we take the first step toward understanding finite sample guarantees of model-free average-reward RL algorithms. Specifically, we consider (i) average-reward TD(λ) with linear

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

function approximation for policy evaluation, and (ii) average-reward tabular Q-learning in the synchronous setting for the control problem.

1.1 Contributions and Summary of Our Techniques

We establish the first finite sample convergence guarantees of average-reward TD(λ) with linear function approximation and average-reward Q-learning in the literature.

TD(λ) Results. We study the average-reward TD(λ) with linear function approximation under a general asynchronous update. We present finite-sample bounds under both constant and diminishing step sizes. With a constant step-size, the iterates converge at an exponential rate to a small cylinder around the set of TD fixed points. With properly chosen diminishing step sizes, the mean-square distance of the iterates to this set converges with an $\tilde{\mathcal{O}}\left(\frac{1}{T}\right)$ rate, and this leads to a sample complexity of $\tilde{\mathcal{O}}\left(\frac{1}{\epsilon^2}\right)$. Our sample complexity bound suggests a trade-off in choosing λ, i.e., the optimal λ should be neither too large nor too small. The dependence on the effective horizon plays a key role in the study of discounted-reward RL algorithms [18, 19]. There is no such effective horizon in average-reward problems, and the spectral gap of an appropriately defined matrix plays a key role instead.

TD(λ) Analysis. A major challenge in the analysis is that the projected Bellman operator is not a contraction under any norm. Moreover, even though the projected Bellman equation can be written as a linear set of equations, they are underdetermined. So existing techniques [13, 14] are not directly applicable. Since the value function is unique up to an additive constant, we have a unique solution of the projected Bellman equation when restricted to an appropriately defined subspace. We exploit this property and work in this subspace and use a quadratic Lyapunov function to obtain finite sample guarantees.

Q-learning Results. We consider a J-step synchronous Q-learning algorithm. We present finite sample error bounds under diminishing step-sizes. The span of a vector is defined to be the difference between the maximum and minimum element. Since the optimal value function, Q^*, is agnostic up to an additive constant, we show that the mean-square of the span of the error $Q_t - Q^*$ converges at a $\mathcal{O}(\frac{1}{t})$ rate. This implies a sample complexity of $\mathcal{O}(\frac{1}{\epsilon^2})$ to find an ϵ-optimal differential Q-function.

Q-learning Analysis. While the average reward Bellman operator is not a contraction under any norm, it is known to be a contraction under the span seminorm. The span seminorm can be interpreted as the ℓ_∞ distance to the space spanned by the all-ones vector. Finite sample bounds for stochastic approximation of ℓ_∞-norm contractive operators. were obtained in [17] by using generalized Moreau envelop as a smooth Lyapunov function. Here, we generalize this approach and introduce a new Lyapunov function to study span seminorm contractive operators. Our Lyapunov function is obtained by applying an infimal convolution with respect to an indicator function to the generalized Moreau envelop used in [17].

1.2 Related Literature

Average-reward MDP: There is an extensive body of literature on average-reward MDPs. Several authors have made early contributions to the average-reward problem [20, 6, 7, 21, 22]. Please see [8] and [23] for the textbook development. There are well known classical dynamic programming algorithms for finding optimal policies such as policy iteration [6] and value iteration [24]. However, these algorithms require knowledge of the state transition probabilities, and are also computationally intractable in large state spaces [25].

Average-reward policy evaluation: Tsitsiklis and Van Roy [9] proved the convergence of an average-reward TD(λ) with linear function approximation, and provided approximation error bounds. The paper did not study finite-sample bounds. Yu and Bertsekas [26] proved the convergence of an average-reward LSPE(λ) and its rate of convergence for constant step size. Both TD(λ) and LSPE(λ) aim to solve the same projected Bellman equation. However, LSTD(λ) uses simulation to construct directly the low-dimensional quantities defining the equation, instead of only the solution itself like TD(λ). Both papers assumed that the set of basis functions are independent of the all-ones vector, which apparently does not hold in the tabular setting. We do not require such an assumption in this paper. Note that both algorithms above are on-policy. Recently, in addition to a convergent off-policy

tabular method presented in [11], Zhang et al. [12] introduced a convergent off-policy linear function approximation algorithm. Both did not study finite-time convergence guarantees.

Average-reward control: The earliest control algorithms were those introduced by Schwartz [27] and Singh [28] without convergence proofs. The first provably convergent algorithms are RVI Q-learning and SSP Q-learning, introduced by Abounadi, Bertsekas, and Borkar [10]. SSP Q-learning and the algorithm introduced later by Gosavi [29] are limited to MDPs with a special state that is recurrent under all stationary policies, whereas RVI Q-learning is convergent for more general MDPs. Recently, Wan et al.[11] introduced an algorithm without a reference function, which is needed in RVI Q-learning, and proved its convergence with the techniques that are a slight generalization of those in [10]. To the best of our knowledge, our paper is the first work in the literature that studies the finite sample guarantees of a general average-reward Q-learning algorithm.

Stochastic approximation: Many RL algorithms can be viewed through the lens of stochastic approximation (SA). There is a well developed asymptotic theory of SA [30, 31, 32]. The ODE method is a dominant approach used in most asymptotic convergence proofs in RL [33]. However, this is a coarse tool, since it is not able to generate insight into an algorithm's sensitivity to noise in the system and step-size choices. Driven by the interest in finite-sample guarantees of RL algorithms, recent years have witnessed a focus shifted from asymptotic analysis to non-asymptotic analysis of SA schemes. For example, a finite-time bound for linear SA is given in [14], which leads to finite time error bounds for TD learning. [16] provides a finite-time analysis of asynchronous nonlinear SA, which yields finite-time bounds on asynchronous Q-learning.

Others: There are other related papers which are beyond the scope of the present paper. For instance, there is a line of work [34, 35, 36] on regret guarantees, which is a different focus compared to our work, for learning in average-reward MDPs. In addition, there are RL methods based on linear programming [37, 38], or learning automata [39, 40].

2 The Average-Reward Problem Setting

We consider an infinite-horizon average-reward MDP described by $(\mathcal{S}, \mathcal{A}, \mathcal{R}, p)$, where $\mathcal{S} = \{1, 2, \cdots, n\}$ is a finite state space, $\mathcal{A} = \{1, 2, \cdots, m\}$ is a finite action space, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to [0, 1]$[①] is the reward function (i.e., $\mathcal{R}(s, a)$ is the immediate reward received upon executing action a while in state s), and $p : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the transition dynamics of the environment (i.e., $p(s'|s, a)$ is the probability of transitioning into state s' upon taking action a in state s). An agent interacts with the environment according to the following protocol: at each time step $t = 0, 1, 2, \cdots$, the agent is in a state $s_t \in \mathcal{S}$ and selects an action $a_t \in \mathcal{A}$, then receives from the environment an immediate reward $\mathcal{R}(s_t, a_t)$ and the next state s_{t+1} which is a sample drawn from $p(\cdot|s_t, a_t)$. The average reward per step of a deterministic stationary policy $\mu : \mathcal{S} \to \mathcal{A}$ starting from state $s \in \mathcal{S}$ is defined as

$$r^{\mu}(s) := \liminf_{T \to \infty} \frac{1}{T} \mathbb{E}\left[\sum_{t=0}^{T-1} \mathcal{R}(s_t, \mu(s_t)) | s_0 = s\right]. \tag{2.1}$$

Let $r^*(s) := \sup_{\mu \in \mathcal{M}} r^{\mu}(s)$, where $\mathcal{M}$ is the set of deterministic stationary policies. A policy $\mu^* \in \mathcal{M}$ is said to be optimal if it satisfies $r^{\mu^*}(s) = r^*(s)$ for all $s \in \mathcal{S}$.

3 Policy Evaluation Algorithm: TD Learning

3.1 Problem Formulation

We consider the problem of evaluating a given policy $\mu \in \mathcal{M}$ when the data is generated by the policy μ. Since the environment is an induced Markov chain, we employ the notation $\mathcal{R}(i) := \mathcal{R}(i, \mu(i))$ for reward, and $P(j|i) := p(j|i, \mu(i))$ for transition probability. We make the following standard assumption to ensure the existence and uniqueness of the stationary distribution, $\pi := [\pi_1, \cdots, \pi_n]^\top$.

[①] We can assume rewards are bounded uniformly such that $|\mathcal{R}(s, a)| \leq r_{\max} < \infty$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, but without much loss of generality and for simplicity of exposition, we assume that $\mathcal{R}(s, a) \in [0, 1]$.

Assumption 1. *The Markov chain associated with P is irreducible and aperiodic*[2].

Notice that π satisfies $\pi^\top P = \pi^\top$ and $\pi_i > 0, \forall i$. Let $\mathbb{E}_\pi[\cdot]$ stand for expectation with respect to π and $D = \text{diag}(\pi_1, \cdots, \pi_n) \in \mathbb{R}^{n\times n}$ denote the diagonal matrix consisting of π. The D-weighted inner product $\langle x, y\rangle_D := x^\top Dy$ and its induced norm, $\|x\|_D = \sqrt{x^\top Dx}$, are helpful.

Under Assumption 1, the average reward in (2.1) satisfies $r^\mu(s) = r(\mu) := \mathbb{E}_\pi[\mathcal{R}(s_t)]$ for all $s \in \mathcal{S}$. A differential value function $v : \mathcal{S} \to \mathbb{R}$ for policy μ satisfies the Bellman equation $Tv = v$, where the Bellman operator $T : \mathbb{R}^n \to \mathbb{R}^n$ is defined by $Tv := \mathcal{R} - r(\mu)e + Pv$. Here, $e \in \mathbb{R}^n$ is the vector with each component equal to 1, and v and $\mathcal{R}$ are viewed as vectors in $\mathbb{R}^n$. Under Assumption 1, it is known that the set of differential value functions takes the form $\{v^\mu + ce | c \in \mathbb{R}\}$, where $v^\mu : \mathcal{S} \to \mathbb{R}$, known as the *basic* differential value function, is given by $v^\mu := \sum_{t=0}^\infty P^t (\mathcal{R} - r(\mu)e)$.

Most modern applications have large state spaces, so due to the curse of dimensionality, exact value function learning may be intractable. To mitigate this, we consider a linear function approximation $V_\theta(i) = \phi(i)^\top\theta$ to differential value functions, where $\phi(i) := [\phi_1(i), \cdots, \phi_d(i)]^\top \in \mathbb{R}^d$ is the feature vector for state $i \in \mathcal{S}$ and $\theta \in \mathbb{R}^d$ is a tunable parameter vector. Here, $\{\phi_k : \mathcal{S} \to \mathbb{R} | k = 1, 2, \cdots, d\}$ is a given set of d basis functions which are viewed as vectors of dimension n, where $d \le n$. With this notation, V_θ can be expressed compactly in the form $V_\theta = \Phi\theta$, where Φ is an $n \times d$ matrix whose k-th column is ϕ_k. We assume that Φ has full column rank; that is, the basis functions $\{\phi_k | k = 1, 2, \cdots, d\}$ are linearly independent. This results in no loss of generality because if some basis function ϕ_k is a linear combination of the others, it can be eliminated without changing the power of the approximation architecture. Additionally, we assume that $\|\phi(i)\|_2 \le 1$ for all $i \in \mathcal{S}$, which can be ensured through feature normalization.

3.2 Average-Reward TD(λ)

Algorithm 1: TD(λ) with linear function approximation

Input : initial guess $\bar{r}_0$ and θ_0, basis functions $\{\phi_k\}_{k=1}^d$, step-size sequence $\{\beta_t\}_{t\in\mathbb{N}}$ and positive constant c_α.

Initialize: $z_{-1} = 0$, $\lambda \in [0, 1)$.

for $t = 0, 1, \ldots$ **do**

- Observe tuple: $O_t = (s_t, \mathcal{R}(s_t), s_{t+1})$
- Get TD error: $\delta_t(\theta_t) = \mathcal{R}(s_t) - \bar{r}_t + \phi(s_{t+1})^\top\theta_t - \phi(s_t)^\top\theta_t$
- Update eligibility trace: $z_t = \lambda z_{t-1} + \phi(s_t)$
- Update average-reward estimate: $\bar{r}_{t+1} = \bar{r}_t + c_\alpha\beta_t(\mathcal{R}(s_t) - \bar{r}_t)$
- Update parameter vector: $\theta_{t+1} = \theta_t + \beta_t\delta_t(\theta_t)z_t$

end

We study the average-reward TD learning with eligibility traces [9], denoted by TD(λ) and parameterized by $\lambda \in [0, 1)$. We consider the Markov chain observation model, where the observed tuples used by TD(λ) are gathered from a single trajectory of the MDP. At every time step t, the algorithm observes one data tuple $O_t := (s_t, \mathcal{R}(s_t), s_{t+1})$ consisting of the current state, the current reward and the next state. Suppose that at some time t, the current value of the parameter vector θ is θ_t, and we have a scalar estimate $\bar{r}_t$ of the average reward $r(\mu)$, we define the TD error $\delta_t(\theta_t)$ corresponding to the transition from s_t to s_{t+1} as $\delta_t(\theta_t) := \mathcal{R}(s_t) - \bar{r}_t + \phi(s_{t+1})^\top\theta_t - \phi(s_t)^\top\theta_t$. TD($\lambda$) updates $\bar{r}_t$ and θ_t as follows:

$$\bar{r}_{t+1} = \bar{r}_t + \alpha_t(\mathcal{R}(s_t) - \bar{r}_t) \text{ and } \theta_{t+1} = \theta_t + \beta_t\delta_t(\theta_t)z_t, \tag{3.1}$$

where α_t and β_t are scalar step sizes, and the vector $z_t := \sum_{k=0}^t \lambda^{t-k}\phi(s_k)$ is called the eligibility trace.

In this work, we focus on the single time-scale variant of TD(λ), that is, we assume that there exists a constant $c_\alpha > 0$ such that $\alpha_t = c_\alpha\beta_t$ for all t. In order to represent TD(λ) in a compact form, we

[2] It is a standard assumption in studying convergence of TD learning algorithms with linear function approximation [41, 9, 13] and guarantees that all states are visited an infinite number of times during an infinitely long trajectory.

construct a process $X_t := (s_t, s_{t+1}, z_t)$. It is easy to see that $\{X_t\}$ is a Markov chain with an infinite state space. If we let $\Theta_t := \begin{bmatrix} \bar{r}_t \\ \theta_t \end{bmatrix}$, the TD($\lambda$) updates (3.1) can be expressed compactly as

$$\Theta_{t+1} = \Theta_t + \beta_t \left[A(X_t)\Theta_t + b(X_t) \right], \tag{3.2}$$

where

$$A(X_t) = \begin{bmatrix} -c_\alpha & 0 \\ -z_t & z_t \left[\phi(s_{t+1})^\top - \phi(s_t)^\top \right] \end{bmatrix} \text{ and } b(X_t) = \begin{bmatrix} c_\alpha \mathcal{R}(s_t) \\ \mathcal{R}(s_t) z_t \end{bmatrix}.$$

Non-uniqueness of TD(λ) limit point. For any $m = 0, 1, \cdots$, the m-step Bellman operator is given by $T^m v = \sum_{t=0}^{m} P^t \left(\mathcal{R} - r(\mu)e \right) + P^{m+1} v$. The asymptotic properties of TD(λ) are closely tied to the λ-weighted version of the m-step Bellman operator. Define the averaged Bellman operator $T^{(\lambda)} v := (1-\lambda) \sum_{m=0}^{\infty} \lambda^m T^{m+1} v$. Note that the set of differential value functions is also the fixed points of $T^{(\lambda)}$; see Lemma 3 of [9] for a proof.

We denote by $A := \mathbb{E}_\pi[A(X_t)]$ and $b := \mathbb{E}_\pi[b(X_t)]$ the steady-state expectations of $A(X_t)$ and $b(X_t)$. Stochastic approximation theory [32] shows that the asymptotic behavior of the sequence $\{\Theta_t\}$ generated by (3.2) is closely linked with the corresponding ordinary differential equation (ODE) $\dot{\Theta}_t = A\Theta_t + b$, and the limit of Θ_t, denoted by $\Theta_\infty = \begin{bmatrix} \bar{r}_\infty \\ \theta_\infty \end{bmatrix}$ if exists, is an equilibrium point of the ODE, i.e.,

$$A\Theta_\infty + b = 0. \tag{3.3}$$

Solving Eq. (3.3) for Θ_∞, we have $r(\mu)$ is the unique solution for $\bar{r}_\infty$, and θ_∞ satisfies the projected Bellman equation

$$\Phi\theta = \Pi_{D,W} T^{(\lambda)} \Phi\theta, \tag{3.4}$$

where $\Pi_{D,W} := \Phi \left(\Phi^\top D \Phi \right)^{-1} \Phi^\top D$ is the projection matrix onto the column space $W := \{\Phi\theta | \theta \in \mathbb{R}^d\}$ of Φ with respect to the norm $\|\cdot\|_D$. It is worth noting that if $e \in W$, then $\Pi_{D,W} T^{(\lambda)}$ has multiple fixed points, since any scalar multiple of e when added to a fixed point of $\Pi_{D,W} T^{(\lambda)}$ would also be a fixed point. For example, in the tabular case where $\Phi = I$, Eq. (3.4) would become $\theta = T^{(\lambda)}\theta$, and it has multiple solutions as we just mentioned above.

3.3 Finite-Time Bounds for TD(λ)

Before we present the finite-time bounds on the performance of TD(λ) with Markovian observation noise, we illustrate the key ideas, which are inspired by [14]. The detailed proof can be found in Appendix A.3.

We study the the drift of an appropriately chosen Lyapunov function to obtain an upper bound on the mean-square error. Denote by $S(\Phi, e) := \text{span}\left(\{\theta | \Phi\theta = e\}\right)$ the linear span of solutions to $\Phi\theta = e$. Since Φ has full rank, $S(\Phi, e) = \{0\}$ if $e \notin W$; otherwise, $S(\Phi, e) = \{c\theta | \Phi\theta = e, c \in \mathbb{R}\}$. Let E be the orthogonal complement of $S(\Phi, e)$. The reason that we define such a set E is that the differential value functions are agnostic to an additive constant. We can also think of E as the set of equivalent classes with the equivalence relation $\sim$ on $\mathbb{R}^d$ defined by $\theta_1 \sim \theta_2$ if and only if $\theta_1 - \theta_2$ is in $S(\Phi, e)$. The following Lemma characterizes the set of TD(λ) fixed points; see Appendix A.1 for the proof.

Lemma 1. *Under Assumption 1, the set of fixed points of the projected Bellman equation (3.4) is*

$$\mathcal{L} := \{\theta^* + c\theta_e \cdot \mathbb{1}\{e \in W\} | c \in \mathbb{R}\},$$

where $\theta^* \in E$ is the unique solution to the equation $\Phi\theta = \Pi_{D,W^*} T^{(\lambda)} \Phi\theta$. Here, $\Pi_{D,W^*}(\cdot)$ is the projection operator onto the subspace $W^* := \{\Phi\theta | \theta \in E\}$ with respect to the norm $\|\cdot\|_D$

Remark 1. *In the case where $e \notin W$, $\mathcal{L} = \{\theta^*\}$, i.e., θ^* is the unique fixed point to the projected Bellman equation (3.4). This is why prior work requires that e does not belong to the subspace W.*

We consider the Lyapunov function $\Phi(\bar{r}, \theta) := \left(\bar{r} - r(\mu)\right)^2 + \|\Pi_{2,E}\left(\theta - \theta^*\right)\|_2^2$. Here, $\Pi_{2,E}$ is the projection matrix onto the subspace E with respect to the 2-norm. Note that $\|\Pi_{2,E}\left(\theta - \theta^*\right)\|_2^2$

measures the distance between θ and $\mathcal{L}$. The following Lemma establishes that the matrix A in (3.3) is negative definite on the subspace $\mathbb{R} \times E$ for a sufficiently large c_α. The proof is presented in Appendix A.2. With this result, we can show that the Lyapunov function Φ has a one-time-step negative drift.

Lemma 2. *Under Assumption 1, we have*

$$\Delta := \min_{\|\theta\|_2=1, \theta \in U^\perp} \theta^\top \Phi^\top D \left(I - P^{(\lambda)}\right) \Phi\theta > 0, \tag{3.5}$$

where $P^{(\lambda)} = (1-\lambda)\sum_{m=0}^{\infty} \lambda^m P^{m+1}$. *Furthermore, when* $c_\alpha \geq \Delta + \sqrt{\frac{1}{\Delta^2(1-\lambda)^4} - \frac{1}{(1-\lambda)^2}}$, *we have*

$$\min_{\|\Theta\|_2=1, \Theta \in \mathbb{R}\times E} -\Theta^\top A \Theta \geq \frac{\Delta}{2}.$$

To handle the Markovian noise, we use the conditioning argument along with the geometric mixing of the underlying Markov chain $\{X_t\}$. Thus, we consider the following definition of the mixing time of a Markov chain.

Definition 1. *Given a positive constant* ϵ, *we define* $\tau(\epsilon) \geq 1$ *to be the mixing time of a Markov chain* $\{X_t\}$ *such that*

$$\|\mathbb{E}\left[A(X_t)|X_0\right] - A\|_2 \leq \epsilon \text{ and } \|\mathbb{E}\left[b(X_t)|X_0\right] - b\|_2 \leq \epsilon, \quad \forall X_0, t \geq \tau(\epsilon)$$

The following lemma establishes that the expectations of $A(X_t)$ and $b(X_t)$ converge to their steady-state values at a geometric rate. See Lemma 6.7 of [42] for a proof.

Lemma 3. *Under Assumption 1, the Markov chain* $\{X_t\}$ *has a geometric mixing time, i.e., given a small positive constant* ϵ, *there exists a constant* $K \geq 1$ *such that* $\tau(\epsilon) \leq K \ln\left(\frac{1}{\epsilon}\right)$.

We now state two finite-time bounds on the performance of TD(λ). Part (a) studies TD(λ) applied with sufficiently small constant step-size, which is common in practice. In this case, the iterate θ_t will never converge to TD(λ) fixed points due to the noise variance, but our result shows that the expected distance to the set of TD(λ) fixed points decreases at an exponential rate below some level that depends on the choice of step-size. Part (b) attains an $\tilde{\mathcal{O}}\left(\frac{1}{T}\right)$ convergence rate to TD(λ) fixed points with a carefully chosen decaying step-size sequence.

Theorem 1. *Consider iterates* $\{(\bar{r}_t, \theta_t)\}$ *generated by Algorithm 1 with Assumption 1 and* $c_\alpha \geq \Delta + \sqrt{\frac{1}{\Delta^2(1-\lambda)^4} - \frac{1}{(1-\lambda)^2}}$. *Let* $\xi_1 := \left(2\sqrt{\bar{r}_0^2 + \|\theta_0\|_2^2} + \sqrt{r(\mu)^2 + \|\theta^*\|_2^2} + 1\right)^2$ *and* $\xi_2 := 228\eta^2\left(\sqrt{r(\mu)^2 + \|\theta^*\|_2^2} + 1\right)^2$, *where* $\eta := \sqrt{c_\alpha^2 + \frac{5}{(1-\lambda)^2}}$.

(a) If $\beta_t = \beta$ *for all* t, *where positive constant* β *is properly chosen such that* $\Delta\beta < 2$ *and* $\beta\tau(\beta) \leq \min\{\frac{1}{4\eta}, \frac{\Delta}{228\eta^2}\}$. *Then, for all* $T \geq \tau(\beta)$, *we have*

$$\mathbb{E}\left[(\bar{r}_T - r(\mu))^2\right] + \mathbb{E}\left[\|\Pi_{2,E}\left(\theta_T - \theta^*\right)\|_2^2\right] \leq \xi_1\left(1 - \frac{\Delta}{2}\beta\right)^{T-\tau(\beta)} + \xi_2\frac{\beta\tau(\beta)}{\Delta}.$$

(b) If $\beta_t = \frac{c_1}{t+c_2}$ *where positive constants* c_1 *and* c_2 *are properly chosen such that* $2 < \Delta c_1 < 2c_2$ *and there exists a smallest positive integer* t^* *such that* $\beta_{0,t^*-1} \leq \frac{1}{2\eta}$, *and for all* $t \geq t^*$, $\beta_{t-\tau(\beta_t),t-1} \leq \min\{\frac{1}{4\eta}, \frac{\Delta}{228\eta^2}\}$. *Then, for all* $T \geq t^*$, *we have*

$$\mathbb{E}\left[(\bar{r}_T - r(\mu))^2\right] + \mathbb{E}\left[\|\Pi_{2,E}\left(\theta_T - \theta^*\right)\|_2^2\right] \leq \xi_1\left(\frac{t^*+c_2}{T+c_2}\right)^{\frac{\Delta c_1}{2}} + \frac{8eKc_1^2\xi_2}{\Delta c_1 - 2}\frac{\ln(T+c_2) - \ln(c_1)}{T+c_2+1}.$$

Therefore, with an appropriate stepsize, the following sample complexity of TD(λ) can be obtained.

Corollary 1. *To find a pair $(\bar{r}, \theta)$ with $\mathbb{E}\left[|\bar{r} - r(\mu)|\right] \leq \epsilon$ and $\mathbb{E}\left[\|\Pi_{2,E}\left(\theta - \theta^*\right)\|_2\right] \leq \epsilon$, Algorithm 1 requires at most the following number of samples:*

$$\tilde{O}\left(\frac{K \log\left(\frac{1}{\Delta}\right)\left(1 + \|\theta^*\|_2^2\right)}{\Delta^4 (1-\lambda)^4 \epsilon^2}\right),$$

where K is the mixing time constant defined in Lemma 3.

Remark 2. *Since Δ defined in (3.5) is a non-decreasing function of λ, the sample complexity in Corollary 1 implies that choosing the optimal λ involves some trade-off between the $1/\Delta$ and $1/(1-\lambda)$.*

4 Control Algorithm: Q-learning

4.1 Problem Formulation

We consider the problem of finding an optimal policy $\mu^* \in \mathcal{M}$ under a standard unichain assumption (e.g. Section 8.4 in [8]).

Assumption 2. *The underlying MDP is called unichain if the induced Markov chain consists of a single recurrent class plus a possibly empty set of transient states for any stationary deterministic policy.*

This assumption guarantees that $r^*(s) = r^*$ for all $s \in \mathcal{S}$, and the solutions of the following Bellman equation, $Q^* : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, are gain-optimal [8]:

$$Q^*(s,a) = \underbrace{\mathcal{R}(s,a) + \textstyle\sum_{s' \in \mathcal{S}} p(s'|s,a) \max_{a' \in \mathcal{A}} Q^*(s',a')}_{H(Q^*)(s,a)} - r^*, \quad \forall s \in \mathcal{S}, a \in \mathcal{A}. \tag{4.1}$$

Note that the optimal policy μ^* can be recovered by computing $\mu^*(s) = \operatorname{argmax}_{a \in \mathcal{A}} Q^*(s,a)$. Next we will use the synchronous Q-learning algorithm to solve (4.1) approximately with a provable convergence guarantee.

4.2 Synchronous Q-learning

Algorithm 2: J-step Synchronous Q-learning Algorithm

Input : initial guess Q_0, step-size sequence $\{\eta_t\}_{t \in \mathbb{N}}$, offset function $f(\cdot)$.
for $t = 0, 1, \ldots$ **do**
  Compute the Q_t-improving policy μ_t.
  for $(s,a) \in \mathcal{S} \times \mathcal{A}$ **do**
    Sample $(s^1, r^1) \sim \mathcal{M}(s,a), (s^2, r^2) \sim \mathcal{M}(s^1, \mu_t(s^1)), \ldots, (s^J, r^J) \sim \mathcal{M}(s^{J-1}, \mu_t(s^{J-1}))$.
    Compute
    $Q_{t+1}(s,a) \leftarrow Q_t(s,a) + \eta_t \left(\sum_{j=1}^J r^j + Q_t(s^J, \mu_t(s^J)) - Q_t(s,a) - f(Q_t)\right).$
  end
end

We use the following stochastic approximation algorithm to solve the Bellman equation (4.1):

$$Q_{t+1} = Q_t + \eta_t \left(\hat{H}(Q_t) - Q_t\right), \tag{4.2}$$

where η_t is some non-negative step-size and $\hat{H}$ is an unbiased estimator of the Bellman operator H given by

$$\hat{H}Q(s,a) = \mathcal{R}(s,a) + \max_{a' \in \mathcal{A}} Q_t(s',a'), \tag{4.3}$$

where $s' \sim P(\cdot|s,a)$ is the next random state.

Sometimes it might be necessary to perform H for J steps before updating Q to ensure convergence. More specifically, let $\mu_Q(s) := \arg\max_a Q(s,a)$ denote the Q-improving policy, and the J-step Bellman operator be defined as

$$\begin{aligned} H^J(Q)(s,a) = \mathcal{R}(s,a) + \mathbb{E}_{s^1 \sim P(\cdot|s,a),\ldots,s^J \sim P(\cdot|s^{J-1},\mu_Q(s^{J-1}))} \big[\\ R(s^1,\mu_Q(s^1)) + \ldots + R(s^{J-1},\pi_Q(s^{J-1})) + Q(\mu^J,\pi_Q(s^J)) \big]. \end{aligned} \tag{4.4}$$

Clearly, the associated Bellman equation $Q = H^J(Q)$ has the same solutions as (4.1) and the corresponding J-step SA algorithm is the same as (4.2) except that using

$$\hat{H}^J Q(s,a) := \mathcal{R}(s,a) + R(s^1,\mu_Q(s^1)) + \ldots + R(s^{J-1},\pi_Q(s^{J-1})) + Q(\mu^J,\pi_Q(s^J)),$$

as the sample Bellman operator, where $s^1 \sim P(\cdot|s,a),\ldots,s^J \sim P(\cdot|s^{J-1},\mu_Q(s^{J-1}))$. Then we arrive at Algorithm 2. Notice that we include an extra offset function to ensure numerical stability (see Section 8.5.5 of [8]). If the offset function $f(\cdot)$ satisfies Assumption 2.2 in [10] and $J = 1$, Algorithm 2 recovers the RVI Q-learning algorithm.

4.3 Finite-Time Analysis

The existing approaches for showing convergence to a fixed point, for e.g. [17], is no longer applicable because the Bellman operator H^J is indifferent to any constant shifting:

$$H^J(Q + ce) = H^J(Q) + ce.$$

And the induced Q-improving policy and its suboptimality gap are the same for any constant shifted Q [8]. Thus it is sensible to view all constant shifts of a Q function, $Q_E := \{Q + ce : c \in \mathbb{R}\}$, as an equivalent class and directly analyze the convergence of those equivalent classes. Towards that end, we define the distance between any Q_1 and Q_2 as the minimum distance between any two members of their associated equivalent classes,

$$\|Q_1 - Q_2\|_{\text{sp}} = \min_{c_1,c_2 \in \mathbb{R}} \|(Q_1 - c_1 e) - (Q_2 - c_2 e)\|. \tag{4.5}$$

We make a span contraction assumption similar to Theorem 8.5.2 in [8].

Assumption 3. *The Bellman operator H^J is a span contraction for some $J \geq 1$, i.e., there exists a $\gamma \in [0,1)$ such that for any Q_1, Q_2*

$$\left\|H^J Q_1 - H^J Q_2\right\|_{\infty,\text{sp}} \leq \gamma \|Q_1 - Q_2\|_{\infty,\text{sp}}.$$

Such an assumption is not restrictive because the aperiodic transformation suggested in Section 8.5.4 of [8] can always be applied in the synchronous setting to ensure its satisfaction (with $J \leq |S|$).

Now we construct an Lyaponov function to analyze the $\{Q_t\}$ convergence. The key insight is that the span norm (4.5) can be interpreted as the infimal convolution with respect to some indicator function

$$\|x\|_{\text{sp}} = (\|\cdot\| \,\square\, \delta_E)(x) := \inf_y f(x-y) + g(y) \quad \forall x, \tag{4.6}$$

where $\delta_E(x) := \begin{cases} 0 & x \in E := \{ce : c \in \mathbb{R}\}, \\ \infty & \text{otherwise.} \end{cases}$

As illustrated by Lemma 4, the inifimal convolution operator has many desirable properties. For example, it is commutative, associative, convexity-preserving and smoothness-preserving. These nice properties allow us to design a Lyaponov function for the equivalent classes by tweaking the usual Lyaponov function M as follows,

$$M_E(x) := \inf_y M(x-y) + \delta_E(y) \equiv M \square \delta_E(x). \tag{4.7}$$

In particular, since Assumption 3 implies only l_∞ contraction, we utilize the smoothed Lyaponov function proposed in [17], $M(Q) := \frac{1}{2}(\|\cdot\|_\infty^2 \,\square\, \|\cdot\|_p^2)$ with $p = 4\log(|S||A|)$. We can show that M_E is a uniform approximation of $\|\cdot\|_{\infty,\text{sp}}$:

$$(1 + (1+1/\gamma)^2/4\sqrt{e})M_E(Q) \leq \tfrac{1}{2}\|Q\|_{\infty,\text{sp}}^2 \leq (1 + (1+1/\gamma)^2/4)M_E(Q) \;\forall Q,$$

and that it is smooth with respect to the span p-norm (4.5):

$$\begin{aligned} M_E(Q_{t+1}-Q_*) \leq & M_E(Q_t-Q_*) + \langle \nabla M_E(Q_t-Q_*), Q_{t+1}-Q_*-(Q_t-Q_*)\rangle \\ & + \tfrac{2\log(|S||A|)}{(1+1/\gamma)^2}\left\|Q_{t+1}-Q_t\right\|_{p,\mathrm{sp}}^2. \end{aligned} \tag{4.8}$$

The γ-contraction assumption leads to a negative drift in (4.8):

$$\begin{aligned} \mathbb{E}\langle \nabla M_E(Q_t-Q_*), Q_{t+1}-Q_*-(Q_t-Q_*)\rangle & \overset{convex}{\leq} M_E(H(Q_t)-Q_*) - M_E(Q_t-Q_*) \\ & \leq -\tfrac{1}{2}(1-\gamma)M_E(Q_t-Q_*). \end{aligned} \tag{4.9}$$

Thus we can derive the following recursive relationship between Q_{t+1} and Q_t:

$$\mathbb{E}[M_E(Q_{t+1}-Q_*)] \leq (1-2\alpha_2\eta_t+\alpha_3\eta_t^2)M_E(Q_t-Q_*)+\alpha_4\eta_t^2, \tag{4.10}$$

where

$$\alpha_2=(1-\gamma)/4, \alpha_3=320\log(|S||A|)/(1-\gamma), \alpha_4=4(J^2+\|Q_*\|_{\infty,\mathrm{sp}}^2)\log|S||A|/(1-\gamma).$$

Clearly, by taking a small enough stepsize η_t, (4.10) implies the convergence of $\{\mathbb{E}[M_E(Q_t-Q_*)]\}_t$. Next we suggest a specific stepsize and calculate its sample complexity.

Theorem 2. *Let $\{Q_t\}$ be generated by Algorithm 2 with a decreasing step-size*

$$\eta_t=\tfrac{4}{1-\gamma}\tfrac{1}{t+K} \text{ where } K=\tfrac{1280\log|S||A|}{(1-\gamma)^2},$$

then we have

$$\mathbb{E}[\frac{1}{2}\left\|Q_t-Q^*\right\|_{\infty,\mathrm{sp}}^2] \leq C\left(\tfrac{\|Q_0-Q_*\|_{\infty,\mathrm{sp}}^2(\log|S||A|)^2}{(1-\gamma)^4t^2}+\tfrac{J^2+\|Q_*\|_{\infty,\mathrm{sp}}^2\log|S||A|}{(1-\gamma)^3t}\right),$$

for some universal constant C. Therefore, it takes at most the following number of samples to find a Q_t with $\mathbb{E}[\|Q_t-Q^*\|_{\infty,\mathrm{sp}}] \leq \epsilon$ for any $\epsilon>0$

$$\max\left\{\mathcal{O}\{\tfrac{\|Q_0-Q_*\|_{\infty,\mathrm{sp}}^2}{(1-\gamma)^2\epsilon}\}, \mathcal{O}\{\tfrac{J^2+\|Q^*\|_{\infty,\mathrm{sp}}^2}{(1-\gamma)^3\epsilon^2}\}\right\} J|S||A|\log(|S||A|).$$

Since γ span contraction assumption implies $\|Q^*\|_{\infty,\mathrm{sp}} \leq \frac{J}{1-\gamma}$, the above sample complexity can be simplified further to

$$\mathcal{O}\{\tfrac{|S||A|J^3}{(1-\gamma)^5\epsilon^2}\},$$

which is similar to the usual complexity of γ-discounted Q-learning algorithm [17, 43].

5 Conclusion

We establish the first finite-sample convergence bounds of (i) average-reward TD(λ) with linear function approximation under Markovian observation noise, and (ii) average-reward tabular Q-learning in the synchronous setting. These RL algorithms can be viewed as stochastic approximation schemes to solve average-reward Bellman equations. However, the Bellman operators are not contractive under any norm. To resolve this difficulty, we construct Lyapunov functions using projection and infimal convolution to analyze the convergence of equivalent classes generated by these algorithms. Our approach is simple and general, so we expect it to have broader applications in other problems.

When analyzing the average-reward Q-learning algorithm, we made a J-step span contraction assumption (Assumption 4), which is not needed for the asymptotic convergence [9]. However, it is unclear if such an assumption is necessary for establishing any finite time convergence bound. Since our results are the first finite sample bounds, a future research direction is on relaxing this assumption.

References

[1] Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.

[2] Prasad Tadepalli, DoKyeong Ok, et al. H-learning: A reinforcement learning method to optimize undiscounted average reward. 1994.

[3] Ilaria Giannoccaro and Pierpaolo Pontrandolfo. Inventory management in supply chains: a reinforcement learning approach. *International Journal of Production Economics*, 78(2):153–161, 2002.

[4] Peter Marbach, Oliver Mihatsch, and John N Tsitsiklis. Call admission control and routing in integrated services networks using neuro-dynamic programming. *IEEE Journal on selected areas in communications*, 18(2):197–208, 2000.

[5] Poj Tangamchit, John M Dolan, and Pradeep K Khosla. The necessity of average rewards in cooperative multirobot learning. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 2, pages 1296–1301. IEEE, 2002.

[6] Ronald A Howard. Dynamic programming and markov processes. 1960.

[7] David Blackwell. Discrete dynamic programming. *The Annals of Mathematical Statistics*, pages 719–726, 1962.

[8] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[9] John N Tsitsiklis and Benjamin Van Roy. Average cost temporal-difference learning. *Automatica*, 35(11):1799–1808, 1999.

[10] Jinane Abounadi, D Bertsekas, and Vivek S Borkar. Learning algorithms for markov decision processes with average cost. *SIAM Journal on Control and Optimization*, 40(3):681–698, 2001.

[11] Yi Wan, Abhishek Naik, and Richard S Sutton. Learning and planning in average-reward markov decision processes. *arXiv preprint arXiv:2006.16318*, 2020.

[12] Shangtong Zhang, Yi Wan, Richard S Sutton, and Shimon Whiteson. Average-reward off-policy policy evaluation with function approximation. *arXiv preprint arXiv:2101.02808*, 2021.

[13] Jalaj Bhandari, Daniel Russo, and Raghav Singal. A finite time analysis of temporal difference learning with linear function approximation. *arXiv preprint arXiv:1806.02450*, 2018.

[14] Rayadurgam Srikant and Lei Ying. Finite-time error bounds for linear stochastic approximation andtd learning. In *Conference on Learning Theory*, pages 2803–2830. PMLR, 2019.

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

[16] Guannan Qu and Adam Wierman. Finite-time analysis of asynchronous stochastic approximation and q-learning. In *Conference on Learning Theory*, pages 3185–3205. PMLR, 2020.

[17] Zaiwei Chen, Siva Theja Maguluri, Sanjay Shakkottai, and Karthikeyan Shanmugam. Finite-sample analysis of stochastic approximation using smooth convex envelopes. *arXiv preprint arXiv:2002.00874*, 2020.

[18] Ashwin Pananjady and Martin J Wainwright. Instance-dependent ℓ_∞-bounds for policy evaluation in tabular reinforcement learning. *IEEE Transactions on Information Theory*, 67(1):566–585, 2020.

[19] Zaiwei Chen, Siva Theja Maguluri, Sanjay Shakkottai, and Karthikeyan Shanmugam. A lyapunov theory for finite-sample guarantees of asynchronous q-learning and td-learning variants. *arXiv preprint arXiv:2102.01567*, 2021.

[20] Dean Gillette. 9. stochastic games with zero stop probabilities. In *Contributions to the Theory of Games (AM-39), Volume III*, pages 179–188. Princeton University Press, 2016.

[21] Barry W Brown. On the iterative method of dynamic programming on a finite space discrete time markov process. *The annals of mathematical statistics*, pages 1279–1285, 1965.

[22] Arthur F Veinott. On finding optimal policies in discrete dynamic programming with no discounting. *The Annals of Mathematical Statistics*, 37(5):1284–1294, 1966.

[23] Dimitri P Bertsekas. Dynamic programming and optimal control 3rd edition, volume ii. *Belmont, MA: Athena Scientific*, 2011.

[24] Douglas J White. Dynamic programming, markov chains, and the method of successive approximations. *Journal of Mathematical Analysis and Applications*, 6(3):373–376, 1963.

[25] Sridhar Mahadevan. Average reward reinforcement learning: Foundations, algorithms, and empirical results. *Machine learning*, 22(1):159–195, 1996.

[26] Huizhen Yu and Dimitri P Bertsekas. Convergence results for some temporal difference methods based on least squares. *IEEE Transactions on Automatic Control*, 54(7):1515–1531, 2009.

[27] Anton Schwartz. A reinforcement learning method for maximizing undiscounted rewards. In *Proceedings of the tenth international conference on machine learning*, volume 298, pages 298–305, 1993.

[28] Satinder P Singh. Reinforcement learning algorithms for average-payoff markovian decision processes. In *AAAI*, volume 94, pages 700–705, 1994.

[29] Abhijit Gosavi. Reinforcement learning for long-run average cost. *European Journal of Operational Research*, 155(3):654–674, 2004.

[30] Harold Kushner and G George Yin. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media, 2003.

[31] Vivek S Borkar. *Stochastic approximation: a dynamical systems viewpoint*, volume 48. Springer, 2009.

[32] Albert Benveniste, Michel Métivier, and Pierre Priouret. *Adaptive algorithms and stochastic approximations*, volume 22. Springer Science & Business Media, 2012.

[33] 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.

[34] Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(4), 2010.

[35] Yasin Abbasi-Yadkori, Peter Bartlett, Kush Bhatia, Nevena Lazic, Csaba Szepesvari, and Gellért Weisz. Politex: Regret bounds for policy iteration using expert prediction. In *International Conference on Machine Learning*, pages 3692–3702. PMLR, 2019.

[36] Chen-Yu Wei, Mehdi Jafarnia Jahromi, Haipeng Luo, Hiteshi Sharma, and Rahul Jain. Model-free reinforcement learning in infinite-horizon average-reward markov decision processes. In *International Conference on Machine Learning*, pages 10170–10180. PMLR, 2020.

[37] Mengdi Wang. Primal-dual π learning: Sample complexity and sublinear run time for ergodic markov decision problems. *arXiv preprint arXiv:1710.06100*, 2017.

[38] Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

[39] R Wheeler and K Narendra. Decentralized learning in finite markov chains. *IEEE Transactions on Automatic Control*, 31(6):519–526, 1986.

[40] Hyeong Soo Chang. Decentralized learning in finite markov chains: revisited. *IEEE Transactions on Automatic Control*, 54(7):1648–1653, 2009.

[41] John N Tsitsiklis and Benjamin Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE transactions on automatic control*, 42(5):674–690, 1997.

[42] Dimitri P Bertsekas and John N Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, 1996.

[43] Martin J Wainwright. Stochastic approximation with cone-contractive operators: Sharp ℓ_∞-bounds for q-learning. *arXiv preprint arXiv:1905.06265*, 2019.

[44] Abhishek Gupta, Rahul Jain, and Peter W Glynn. An empirical algorithm for relative value iteration for average-cost mdps. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 5079–5084. IEEE, 2015.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Yes
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]