

# On Convergence of Gradient Expected Sarsa( $\lambda$ )

Long Yang<sup>1</sup>, Gang Zheng<sup>1</sup>, Yu Zhang<sup>1</sup>, Qian Zheng<sup>2</sup>, Pengfei Li<sup>1</sup>, Gang Pan<sup>1\*</sup>

<sup>1</sup>College of Computer Science and Technology, Zhejiang University, China.

<sup>2</sup>School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore.

<sup>1</sup>{yanglong, gang\_zheng, hzzhangyu, pfl, gpan}@zju.edu.cn; <sup>2</sup>zhengqian@ntu.edu.sg

## Abstract

We study the convergence of Expected Sarsa( $\lambda$ ) with function approximation. We show that with off-line estimate (multi-step bootstrapping) to Expected Sarsa( $\lambda$ ) is unstable for off-policy learning. Furthermore, based on convex-concave saddle-point framework, we propose a convergent Gradient Expected Sarsa( $\lambda$ ) (GES( $\lambda$ )) algorithm. The theoretical analysis shows that the proposed GES( $\lambda$ ) converges to the optimal solution at a linear convergence rate under true gradient setting. Furthermore, we develop a Lyapunov function technique to investigate how the step-size influences finite-time performance of GES( $\lambda$ ). Additionally, such a technique of Lyapunov function can be potentially generalized to other gradient temporal difference algorithms. Finally, our experiments verify the effectiveness of our GES( $\lambda$ ). For the details of proof, please refer to <https://arxiv.org/pdf/2012.07199.pdf>.

## Introduction

Tabular Expected Sarsa( $\lambda$ ) (with importance sampling) is one of the widely used methods for off-policy evaluation in reinforcement learning (RL), whose goal is to estimate the value function of a given target policy via the data that is generated from a behavior policy. Due to the high-dimensional state space, instead of tabular learning, a standard approach is to estimate the value function with a linear function (Sutton and Barto 2018). There is very little literature to study Expected Sarsa( $\lambda$ ) with function approximation for off-policy learning. To our best knowledge, Sutton and Barto (2018) (section 12.9) firstly extend *off-line* estimate (multi-step bootstrapping) to Expected Sarsa( $\lambda$ ) with linear function approximation.

Unfortunately, as pointed out by Sutton and Barto (2018) intuitively, their off-line approach may be unstable, i.e., their way to extend Expected Sarsa( $\lambda$ ) with linear function approximation still lacks a provable convergence guarantee, which is undesirable for RL. It is critical to find the inherent essence of the above unstable appearance in Expected Sarsa( $\lambda$ ), which not only makes a complement for existing off-policy learning methods but also provides

some inspirations to design a stable algorithm. Thus, extending Expected Sarsa( $\lambda$ ) with linear function approximation for off-policy evaluation is a fundamental theoretical topic in RL, including: 1) how to character the instability of off-line Expected Sarsa( $\lambda$ ) with linear function approximation; 2) how to derive a convergent algorithm; 3) what convergence rate does Expected Sarsa( $\lambda$ ) with linear function approximation can reach. We focus on these questions in this paper.

**Our Main Works** To address the above problems, we propose Theorem 1, which character a sufficient and necessary condition that presents stability criteria of off-line update Expected Sarsa( $\lambda$ ) with linear function approximation. Theorem 1 requires the *key* matrix (that has been defined in (10)) keeps the negative real components. Unfortunately, due to the discrepancy between behavior policy and target policy, off-line Expected Sarsa( $\lambda$ ) that is suggested by Sutton and Barto (2018) may not satisfy the condition appears in Theorem 1, i.e., their scheme maybe unstable. Then, we use a classic counterexample to verify the above instability lies in the off-line update Expected Sarsa( $\lambda$ ) with linear function approximation, see Example 1.

Furthermore, to get a stable algorithm, we derive an *on-line* gradient Expected Sarsa( $\lambda$ ) (GES( $\lambda$ )) algorithm. Theorem 2 shows that the proposed GES( $\lambda$ ) learns the optimal solution at a linear convergence rate under a true gradient. Finally, inspired by Srikant and Ying (2019), Wang et al. (2019), Gupta et al. (2019), we develop a Lyapunov function technique to establish Theorem 3, which illustrates the relationship between the finite-time performance of GES( $\lambda$ ) and step-size. Result shows that the upper-bounded error consists of two different parts: the first error depends on both step-size and the size of samples, and such error decays geometrically as the samples increase; while the second error is only determined by the step-size and it is independent of samples. Additionally, the technique of proving Theorem 3 can be potentially generalized to other GTD algorithms.

## Notations

We use  $\text{Spec}(A)$  to denote the eigenvalues of the matrix  $A \in \mathbb{C}^{p \times p}$ , i.e.,  $\text{Spec}(A) = \{\lambda_1, \dots, \lambda_p\}$ , where  $\lambda_i$  is the root of the characteristic equation  $p(\lambda) = \det(A - \lambda I)$ . We use  $\mathbb{C}_-$  to denote the collection that contains the complex numbers with negative real components, i.e.,

$$\mathbb{C}_- =: \{c \in \mathbb{C}; \text{Re}(c) < 0\}.$$

\*Corresponding author.

Let  $\lambda_{\min}(A)$  and  $\lambda_{\max}(A)$  be the minimum and maximum eigenvalue of the matrix  $A$  correspondingly. We use  $\|A\|_{\text{op}}$  to denote the operator norm of matrix  $A$ ; furthermore, if  $A$  is a symmetric real matrix, then  $\|A\|_{\text{op}} = \max_{1 \leq i \leq p} \{|\lambda_i|\}$ .  $\kappa(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$  is the condition number of matrix  $A$ . We use  $A \succ 0$  to denote a positive definite matrix  $A$ .

For a function  $f(x) : \mathbb{R}^p \rightarrow \mathbb{R}$ , let  $\nabla^2 f(x)$  denote its Hessian matrix, and its convex conjugate function  $f^*(y) : \mathbb{R}^d \rightarrow \mathbb{R}$  is defined as  $f^*(y) = \sup_{x \in \mathbb{R}^p} \{y^\top x - f(x)\}$ .

**Fact 1** ((Rockafellar 1970; Kakade et al. 2009)). *Let  $f(\cdot)$  be  $\alpha$ -strongly convex and  $\beta$ -smooth, i.e.,  $f(\cdot) - \frac{\alpha}{2}\|\cdot\|_2^2$  is convex and  $\|f(u) - f(v)\| \leq \beta\|u - v\|$ . If  $0 \leq \alpha \leq \beta$ , then the following fact holds,*

- (I)  $f^*$  is  $\frac{1}{\alpha}$ -smooth and  $\frac{1}{\beta}$ -strongly convex.
- (II)  $\nabla f^* = (\nabla f)^{-1}$  and  $\nabla f = (\nabla f^*)^{-1}$ .

## Preliminary

Reinforcement learning (RL) is formalized as Markov decision processes (MDP) which considers the tuple  $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$ ;  $\mathcal{S}$  is the space of states,  $\mathcal{A}$  is the space of actions;  $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ ,  $p_{ss'}^a = P(S_t = s' | S_{t-1} = s, A_{t-1} = a)$  is the probability of state transition from  $s$  to  $s'$  under playing the action  $a$ ;  $R(\cdot, \cdot) : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^1$  is the expected reward function;  $\gamma \in (0, 1)$  is the discount factor.

The policy  $\pi$  is a probability distribution on  $\mathcal{S} \times \mathcal{A}$ , we use  $\pi(a|s)$  to denote the probability of playing  $a$  under the state  $s$ . Let  $\{S_t, A_t, R_{t+1}\}_{t \geq 0}$  be generated by a given policy  $\pi$ , its *state-action value function*

$$q^\pi(s, a) = \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \gamma^t R_{t+1} | S_0 = s, A_0 = a \right],$$

where  $\mathbb{E}_\pi[\cdot|\cdot]$  is the conditional expectation on the actions selected according to  $\pi$ . Let  $\mathcal{B}^\pi : \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|} \rightarrow \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}$  be the *Bellman operator*:

$$\mathcal{B}^\pi : q \mapsto R^\pi + \gamma P^\pi q, \quad (1)$$

where  $P^\pi \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ ,  $R^\pi \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}$ , their corresponding elements are:

$$[P^\pi]_{s,s'} = \sum_{a \in \mathcal{A}} \pi(a|s) p_{ss'}^a, [R^\pi]_{s,a} = R(s, a).$$

It is well-known that  $q^\pi$  is the unique fixed point of  $\mathcal{B}^\pi$ , i.e.,  $\mathcal{B}^\pi q^\pi = q^\pi$ , which is known as Bellman equation.

**Off-Policy Evaluation** Let's consider the following trajectory  $\mathcal{T}$  generated by a policy  $\mu$ :

$$\mathcal{T} = \{S_0 = s, A_0 = a, \dots, S_t, A_t, R_{t+1}, \dots\},$$

where  $A_t \sim \mu(\cdot|S_t)$ ,  $S_{t+1} \sim P(\cdot|S_t, A_t)$ . In RL, the task of off-policy evaluation is to estimate the value function of the target policy  $\pi$  via the data that is generated by another policy  $\mu$  (that is called *behavior policy*), where  $\mu \neq \pi$ .

**Assumption 1.** *The Markov chain induced by behavior policy  $\mu$  is ergodic, i.e., there exists a stationary distribution  $\xi(\cdot, \cdot)$  over  $\mathcal{S} \times \mathcal{A}$ : for  $\forall (S_0, A_0) \in \mathcal{S} \times \mathcal{A}$ ,*

$$\frac{1}{n} \sum_{k=1}^n P(S_k = s, A_k = a | S_0, A_0) \xrightarrow{n \rightarrow \infty} \xi(s, a) > 0. \quad (2)$$

The ergodicity of behavior policy  $\mu$  is a standard assumption in off-policy learning (Bertsekas 2012), and it implies each-action pair can be visited under this behavior policy  $\mu$ . In this paper, we use  $\Xi$  to denote a diagonal matrix whose diagonal element is  $\xi(s, a)$ , i.e.,

$$\Xi = \text{diag}\{\dots, \xi(s, a), \dots\}.$$

**Temporal Difference (TD) Learning** TD learning updates value function as follows,  $\forall t \geq 0$ ,

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha_t \delta_t, \quad (3)$$

where  $Q(\cdot, \cdot)$  is an estimate of state-action value function,  $\alpha_t$  is step-size and  $\delta_t$  is TD error. Let  $Q_t = Q(S_t, A_t)$ , if  $\delta_t$  is expected TD error:

$$\delta_t^{\text{ES}} = R_{t+1} + \gamma \sum_{a \in \mathcal{A}} \pi(a|S_{t+1}) Q(S_{t+1}, a) - Q_t, \quad (4)$$

then update (3) is Expected Sarsa.

**Expected Sarsa**( $\lambda$ ) Sutton and Barto (2018)<sup>1</sup> propose a multi-step TD learning that extends Expected Sarsa to  $\lambda$ -return version: for each  $t \geq 0$ ,

$$G_t^\lambda = Q_t + \sum_{k=t}^{\infty} (\gamma \lambda)^{k-t} \left( \prod_{i=t+1}^k \frac{\pi(A_i|S_i)}{\mu(A_i|S_i)} \right) \delta_k^{\text{ES}}, \quad (5)$$

where  $\delta_k^{\text{ES}}$  is expected TD error. For the convenience, we set  $\prod_{i=t}^k \rho_i = \prod_{i=t}^k \frac{\pi(A_i|S_i)}{\mu(A_i|S_i)} = \rho_{t:k}$ , and  $\rho_{t:t+1} = 1$ .

Finally, we introduce  $\lambda$ -operator  $\mathcal{B}_\lambda^\pi$  that is a high level view of iteration (5):

$$\mathcal{B}_\lambda^\pi : q \mapsto q + \mathbb{E}_\mu \left[ \sum_{k=t}^{\infty} (\lambda \gamma)^{k-t} \delta_k^{\text{ES}} \rho_{t+1:k} \right] \quad (6)$$

$$= q + (I - \lambda \gamma P^\pi)^{-1} (\mathcal{B}^\pi q - q), \quad (7)$$

where  $\mathcal{B}^\pi$  is defined in (1). For the limitation of space, we provide the derivation of (7) from (6) in Appendix A.2.

## Linear Function Approximation

TD learning (3) requires a very huge table to store the estimate value function  $Q(\cdot, \cdot)$  when  $|\mathcal{S}|$  is very large, which implies tabular TD learning is considerably expensive for high-dimensional RL. We often use a parametric function  $Q_\theta(\cdot, \cdot)$  to approximate  $q^\pi(s, a)$ , i.e.,

$$q^\pi(s, a) \approx \phi^\top(s, a) \theta =: Q_\theta(s, a),$$

where  $\theta \in \mathbb{R}^p$  is the parameter that needs to be learned,  $\phi(s, a) = (\varphi_1(s, a), \varphi_2(s, a), \dots, \varphi_p(s, a))^\top$ , and each  $\varphi_i : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ . Furthermore,  $Q_\theta$  can be rewritten as a version of matrix  $Q_\theta = \Phi \theta \approx q^\pi$ , where  $\Phi$  is a matrix whose rows are the state-action feature vectors  $\phi^\top(s, a)$ . In this paper, we mainly consider extending  $\lambda$ -return of Expected Sarsa (5) with linear function approximation.

<sup>1</sup>It is noteworthy that the  $\lambda$ -return version of Expected Sarsa appears in section 12.9 of (Sutton and Barto 2018) is limited in the case of linear function case, Eq.(5) extends it to be a general case.

## Off-Line Gradient Expected Sarsa( $\lambda$ )

In this section, we use a counterexample to show the way to extend Expected Sarsa( $\lambda$ ) with linear function approximation via off-line estimate is unstable for off-policy learning.

**Off-Line Update** Sutton and Barto (2018) provide a way to extend (5) with linear function approximation as follows,

$$\begin{aligned}\theta_{t+1} &= \theta_t + \alpha_t (G_t^\lambda - Q_\theta(S_t, A_t)) \nabla Q_\theta(S_t, A_t) \\ &= \theta_t + \alpha_t \left( \sum_{k=t}^{\infty} (\gamma\lambda)^{k-t} \delta_{k,\theta}^{\text{ES}} \rho_{t+1:k} \right) \phi_t,\end{aligned}\quad (8)$$

where  $\alpha_t$  is step-size,  $\phi_t =: \phi(S_t, A_t)$ , and

$$\delta_{k,\theta}^{\text{ES}} = R_{k+1} + \gamma \theta_t^\top \mathbb{E}_\pi[\phi(S_{k+1}, \cdot)] - \theta_t^\top \phi_k.$$

Furthermore, we can rewrite the expected parameter in (8):

$$\mathbb{E}_\mu[\theta_{t+1}] = \theta_t + \alpha_t (A\theta_t + b), \quad (9)$$

where

$$\begin{aligned}A &= \mathbb{E}_\mu \left( \sum_{k=t}^{\infty} (\gamma\lambda)^{k-t} \rho_{t+1:k} \phi_t (\gamma \mathbb{E}_\pi[\phi(S_k, \cdot)] - \phi_k)^\top \right) \\ &= \Phi^\top \Xi (I - \gamma\lambda P^\pi)^{-1} (\gamma P^\pi - I) \Phi,\end{aligned}\quad (10)$$

$$\begin{aligned}b &= \mathbb{E}_\mu \left( \sum_{k=t}^{\infty} (\gamma\lambda)^{k-t} \rho_{t+1:k} \phi_t R_{k+1} \right) \\ &= \Phi^\top \Xi (I - \gamma\lambda P^\pi)^{-1} R.\end{aligned}\quad (11)$$

If  $\theta_t$  (9) converges to a certain point  $\theta^*$ , then  $\theta^*$  satisfies the following linear system:

$$A\theta^* + b = 0. \quad (12)$$

Such  $\theta^*$  satisfies (12) is called *TD-fixed point*.

**Stability Criteria** According to Sutton et al. (2016); and Ghosh and Bellemare (2020), we formulate the stability of the iteration (9) as the next definition.

**Definition 1** (Stability). *Update rule (9) is stable if  $\theta_k$  converges to the point  $\theta^*$  satisfies (12) for any initial  $\theta_0$ .*

**Theorem 1** (Stability Criteria). *Under Assumption 1, the off-line update (9) is stable if and only if the eigenvalues of the matrix  $A$  (10) have negative real components, i.e.,*

$$\text{Spec}(A) \subset \mathbb{C}_-. \quad (13)$$

We provide its proof in Appendix B. Theorem 1 provides a sufficient and necessary condition (13) that guarantees the stability of iteration (8). Unfortunately, for off-policy learning, the matrix  $A$  (10) can not guarantee the condition (13) holds, which implies the iteration (8) may be divergent and unstable. Now, we use the following example (Touati et al. 2018) to illustrate the instability lies in the iteration (8).

**Example 1.** *For the MDP in Figure 1, we assign the features  $\{(1, 0)^\top, (2, 0)^\top, (0, 1)^\top, (0, 2)^\top\}$  to the state-action pairs  $\{(s_1, \text{right}), (s_2, \text{right}), (s_1, \text{left}), (s_2, \text{left})\}$ . From the dynamic transition shown in Figure 1, we have*

$$P^\pi = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \Phi = \begin{pmatrix} 1 & 0 \\ 2 & 0 \\ 0 & 1 \\ 0 & 2 \end{pmatrix}, \Xi = \frac{1}{2} I_{4 \times 4}.$$

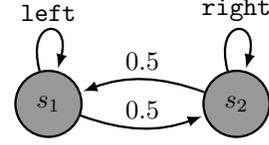


Figure 1: Counterexample, Two-State MDP: behavior policy  $\mu(\text{right}|\cdot) = 0.5$  and target policy  $\pi(\text{right}|\cdot) = 1$ .

Then, according to (10), we have

$$A = \Phi^\top \Xi (I - \gamma\lambda P^\pi)^{-1} (\gamma P^\pi - I) \Phi = \begin{pmatrix} \frac{6\gamma - \gamma\lambda - 5}{2(1 - \gamma\lambda)} & 0 \\ \frac{3\gamma}{2} & -\frac{5}{2} \end{pmatrix},$$

and the eigenvalues of  $A$  are:  $\frac{6\gamma - \gamma\lambda - 5}{2(1 - \gamma\lambda)}$  and  $\frac{5}{2}$ . For any initial  $\theta_0 = (\theta_{0,1}, \theta_{0,2})^\top$ , let  $\mathbb{E}[\theta_{t+1}] =: (\theta_{t+1,1}, \theta_{t+1,2})^\top$  be the expectation of iteration (8), according to (9), the first component of  $\mathbb{E}[\theta_{t+1}|\theta_t]$  is:  $\theta_{t+1,1} = \theta_{0,1} \prod_{i=0}^t \left( 1 + \alpha_i \frac{6\gamma - \gamma\lambda - 5}{2(1 - \gamma\lambda)} \right)$ . For any  $\lambda \in (0, 1)$ , if  $\gamma \in (\frac{5}{6-\lambda}, 1)$ , then  $\frac{6\gamma - \gamma\lambda - 5}{2(1 - \gamma\lambda)}$  is a positive scalar, which implies  $A$  can not be a negative matrix. Furthermore, if step size  $\alpha_t : \sum_{i \geq 0} \alpha_t = \infty$ , we have <sup>2</sup>

$$|\theta_{t+1,1}| = |\theta_{0,1}| \prod_{i=0}^t \left( 1 + \alpha_i \frac{6\gamma - \gamma\lambda - 5}{2(1 - \gamma\lambda)} \right) \rightarrow +\infty, \quad (14)$$

which implies the way (8) to extend Expected Sarsa( $\lambda$ ) with linear function approximation via off-line estimate is unstable for off-policy learning.

## On-Line Gradient Expected Sarsa( $\lambda$ )

The above discussion of the instability for off-policy learning shows that we should abandon the off-line update (8). In this section, we provide a convergent on-line algorithm: Gradient Expected Sarsa( $\lambda$ ) (GES( $\lambda$ )), which is based on the popular TD fixed point method.

The TD fixed point method (Sutton et al. 2009a; Bertsekas 2011; Dann et al. 2014) is widely used for policy evaluation and it focuses on finding the value function satisfies

$$\Phi\theta = \Pi \mathcal{B}_\lambda^\pi \Phi\theta, \quad (15)$$

where  $\Pi = \Phi(\Phi^\top \Xi \Phi)^{-1} \Phi^\top \Xi$ . It has been shown that if the *projected Bellman operator*  $\Pi \mathcal{B}_\lambda^\pi$  has a fixed point  $\theta^*$ , then it is unique (Lagoudakis and Parr 2003; Bertsekas 2011), and such a fixed-point  $\theta^*$  also satisfies the linear system (12).

Instead of using the method of value iteration according to the projected Bellman operator  $\Pi \mathcal{B}_\lambda^\pi$ , we derive the algorithm on the mean square projected Bellman equation (MSPBE) (Sutton et al. 2009a) as follows,

$$\begin{aligned}\min_{\theta} \text{MSPBE}(\theta, \lambda) &=: \min_{\theta} \frac{1}{2} \|\Phi\theta - \Pi \mathcal{B}_\lambda^\pi(\Phi\theta)\|_{\Xi}^2 \\ &= \min_{\theta} \frac{1}{2} \|A\theta + b\|_{M^{-1}}^2,\end{aligned}\quad (16)$$

<sup>2</sup>Eq.(14) is a direct result of the following conclusion that could be found in any calculus textbook. Let  $p_i = 1 + a_i$ , where  $a_i > 0$ , if  $\sum_{i=1}^{\infty} a_i = +\infty$ , then  $\prod_{i=1}^{\infty} p_i = \prod_{i=1}^{\infty} (1 + a_i) = +\infty$ .

---

**Algorithm 1** Gradient Expected Sarsa( $\lambda$ ) (GES( $\lambda$ ))

---

1: **Initialization:**  $\omega_0 = 0, \theta_0 = 0, \alpha_0 > 0, \beta_0 > 0, T \in \mathbb{N}$ .  
2:  $e_{-1} = 0$   
3: **for**  $t = 0$  **to**  $T$  **do**  
4:   Observe  $\{S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1}\} \sim \mu$   
5:    $\rho_t = \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)}$   
6:    $e_t = \lambda\gamma\rho_t e_{t-1} + \phi_t$   
7:    $\delta_t = R_{t+1} + \gamma\theta_t^\top \mathbb{E}_\pi \phi(S_{t+1}, \cdot) - \theta_t^\top \phi_t$   
8:    $\omega_{t+1} = \omega_t + \beta_t(e_t \delta_t - \phi_t \phi_t^\top \omega_t)$   
9:    $\theta_{t+1} = \theta_t - \alpha_t(\gamma \mathbb{E}_\pi[\phi(S_{t+1}, \cdot)] - \phi_t) e_t^\top \omega_t$   
10: **end for**  
11: **Output:**  $\{\theta_t, \omega_t\}_{t=1}^T$

---

where  $\|x\|_\Xi = x^\top \Xi x$  is a weighted norm, and  $M = \Phi^\top \Xi \Phi$ . We provide the derivation of (16) in Appendix C.1.

Since the computational complexity of the invertible matrix  $M^{-1}$  is very large, it is too expensive to use stochastic gradient method to solve the problem (16) directly. Let

$$g(\omega) = \frac{1}{2} \|\omega\|_M^2 - b^\top \omega \quad (17)$$

$$\Psi(\theta, \omega) = (A\theta + b)^\top \omega - \frac{1}{2} \|\omega\|_M^2 = \theta^\top A\omega - g(\omega).$$

According to Liu et al. (2015), the original problem (16) is equivalent to the convex-concave saddle-point problem

$$\min_{\theta} \max_{\omega} \{\Psi(\theta, \omega)\}. \quad (18)$$

**Proposition 1.** *If  $(\theta^*, \omega^*)$  is the solution of problem (18), then  $\theta^*$  is the solution of original problem (16), i.e.,*

$$\theta^* = \arg \min_{\theta} \text{MSPBE}(\theta, \lambda).$$

We provide the proof of Proposition 1 in Appendix C.2. Proposition 1 illustrates that the solution of (16) is contained in the problem (18). Gradient update is a natural way to solve problem (18) (ascending in  $\omega$  and descending in  $\theta$ ):

$$\omega_{t+1} = \omega_t + \beta_t(A\theta_t + b - M\omega_t), \quad (19)$$

$$\theta_{t+1} = \theta_t - \alpha_t A^\top \omega_t, \quad (20)$$

where  $\alpha_t, \beta_t$  is step-size,  $t \geq 0$ .

However, since  $A, b$ , and  $M$  are versions of expectations, we can not get the transition probability in practice. A practical way is to find the unbiased estimators of them. Let  $e_0 = 0, \rho_t = \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)}, e_t = \lambda\gamma\rho_t e_{t-1} + \phi_t, \hat{b}_t = R_{t+1} e_t, \hat{A}_t = e_t(\gamma \mathbb{E}_\pi[\phi(S_{t+1}, \cdot)] - \phi_t)^\top, \hat{M}_t = \phi_t \phi_t^\top$ . According to the Theorem 9 of (Maei 2011), we have

$$\mathbb{E}_\mu[\hat{A}_t] = A, \mathbb{E}_\mu[\hat{b}_t] = b, \mathbb{E}_\mu[\hat{M}_t] = M. \quad (21)$$

Replacing the expectations in (19) and (20) by corresponding unbiased estimates, we define the stochastic on-line implementation of (19) and (20) as follows,

$$\omega_{t+1} = \omega_t + \beta_t(\hat{A}_t \theta_t + \hat{b}_t - \hat{M}_t \omega_t), \quad (22)$$

$$\theta_{t+1} = \theta_t - \alpha_t \hat{A}_t^\top \omega_t. \quad (23)$$

We provide more details in Algorithm 1.

## Finite-Time Performance Analysis

In this section, we mainly focus on the finite-time performance of GES( $\lambda$ ). Theorem 2 shows the proposed GES( $\lambda$ ) with a true gradient can converge at a linear rate under a concrete step-size. Furthermore, to investigate how the step-size influences finite-time performance, we establish Theorem 3. We develop a technique of Lyapunov function (Gupta et al. 2019) to prove Theorem 3, before we present the main result, we provide the motivation and some necessary details of Lyapunov function.

Throughout this paper, we make two additional standard assumptions, which are widely used in reinforcement learning (Wang et al. 2017; Bhandari et al. 2018; Xu et al. 2019).

**Assumption 2** (Boundedness of Feature Map, Reward). *The features  $\{\phi_t\}_{t \geq 0}$  is uniformly bounded by  $\phi_{\max}$ . The reward function is uniformly bounded by  $R_{\max}$ . The importance sampling  $\rho_t$  is uniformly bounded by  $\rho_{\max}$ .*

**Assumption 3** (Solvability of Problem). *The matrix  $A$  is non-singular and  $\text{rank}(\Phi) = p$ .*

As claimed by Xu et al., (2019), Assumption 2 can be ensured by normalizing the feature maps  $\{\phi_t\}_{t \geq 1}$  and when  $\mu(\cdot|s)$  is non-degenerate for all  $s \in \mathcal{S}$ . Besides, Assumption 2 implies the boundedness of the estimators  $\hat{A}_t, \hat{M}_t$  and  $\hat{b}_t$ . For the limitation of space, we provide more details and discussions in Remark 5 (see Appendix D.1).

Assumption 3 requires the non-singularity of the matrix  $A$ , which implies the optimal parameter  $\theta^* = -A^{-1}b$  is well defined. The feature matrix  $\Phi$  has linearly independent columns implies the matrix  $M$  is non-singular.

### Linear Convergence Rate under True Gradient

We consider the first-order optimality condition of the problem (18), i.e., the optimal solution  $(\theta^*, \omega^*)$  satisfies

$$\begin{cases} \nabla_{\theta} \Psi(\theta^*, \omega^*) &= A^\top \omega^* = 0, \\ \nabla_{\omega} \Psi(\theta^*, \omega^*) &= -\nabla g(\omega^*) + A\theta^* = 0. \end{cases} \quad (24)$$

According to the Fact 1 and the condition (24), we have

$$\omega^* = (\nabla g)^{-1}(A\theta^*) = \nabla g^*(A\theta^*),$$

which implies  $\omega^*$  can be represented by  $\theta^*$ , thus, we mainly focus on the performance of  $\{\theta_t\}_{t \geq 1}$ .

**Theorem 2.**  *$\{(\theta_t, \omega_t)\}_{t \geq 0}$  is generated by (19)-(20). Let  $\Delta_{\theta_t} = \|\theta_t - \theta^*\|_2^2, \Delta_{\omega_t} = \|\omega_t - \nabla g^*(A\theta_t)\|_2^2, \nu = \frac{2\kappa^2(A)\kappa(M)\lambda_{\max}(A)}{\lambda_{\min}(M)}$ , and  $D_t = \nu\Delta_{\theta_t} + \Delta_{\omega_t}$ . If we choose*

*step-size  $\alpha = \frac{\lambda_{\min}(M)}{(\lambda_{\max}(M) + \lambda_{\min}(M)) \left( \frac{\lambda_{\max}^2(A)}{\lambda_{\min}^2(M)} + \nu\lambda_{\max}(A) \right)}, \beta = \frac{2}{\lambda_{\max}(M) + \lambda_{\min}(M)}$ , under Assumption 1-3, we have*

$$\mathbb{E}[D_{t+1}] \leq \left(1 - \frac{1}{12\kappa^3(M)\kappa^4(A)}\right) \mathbb{E}[D_t]. \quad (25)$$

Furthermore, we have

$$\mathbb{E}[\|\theta_t - \theta^*\|_2^2] \leq \frac{1}{\nu} \left(1 - \frac{1}{12\kappa^3(M)\kappa^4(A)}\right)^t \mathbb{E}[D_0]. \quad (26)$$

**Remark 1.** We provide its proof in Appendix C.3. Recall the fact  $D_t = \nu \Delta_{\theta_t} + \Delta_{\omega_t} \geq \nu \Delta_{\theta_t}$ , which implies the inequality

$$\mathbb{E}[\|\theta_t - \theta^*\|_2^2] \leq \frac{\mathbb{E}[D_t]}{\nu} \stackrel{(25)}{\leq} \frac{1}{\nu} \left(1 - \frac{1}{12 \kappa^3(M) \kappa^4(A)}\right)^t \mathbb{E}[D_0].$$

The term  $(1 - \frac{1}{12 \kappa^3(M) \kappa^4(A)}) \in (0, 1)$  implies  $\text{GES}(\lambda)$  produces the sequence  $\{\theta_t\}_{t \geq 0}$  converges to the optimal solution at a linear convergence rate. After some simple algebra, with a computational cost of

$$\mathcal{O}\left(\max\left\{1, \frac{\lambda_{\max}(A)}{\lambda_{\max}(M)\nu}\right\} \left(1 - \frac{1}{12 \kappa^3(M) \kappa^4(A)}\right) \log \frac{1}{\delta}\right),$$

the output of Algorithm 1 closes to  $(\theta^*, \omega^*)$  as follows,

$$\mathbb{E}[\|\theta_t - \theta^*\|_2^2] \leq \delta^2, \quad \mathbb{E}[\|\omega_t - \omega^*\|_2^2] \leq \delta^2.$$

**Remark 2.** Theorem 2 provides a concrete step-size that depends on the some unknown parameters. As suggested by Du et al. (2017), Touati et al. (2018), and Voloshin et al. (2019), we can use Monte Carlo method to estimate the unknown parameters.

### A Further Analysis of Lyapunov Function for Stochastic Gradient

In this section, we propose Theorem 3 that illustrates a relationship between the performance of  $\text{GES}(\lambda)$  and step-size.

The proof of Theorem 3 involves a novel Lyapunov function technique, we start by presenting the motivation behind such Lyapunov function. Let

$$H = -A^\top M^{-1}A, L = 2A.$$

Let  $Q_1$  and  $Q_2$  be the solutions of the following equations

$$\begin{cases} -H^\top Q_1 - Q_1 H &= I \\ M^\top Q_2 + Q_2 M &= I. \end{cases} \quad (27)$$

Since both  $M$  and  $-H$  are Hurwitz matrix, the solution of the linear system (27) always exists (Lakshminarayanan et al. 2018; Srikant and Ying 2019). Let  $Q$  be a matrix as follows,

$$Q = \frac{1}{p_1 + p_2} \begin{pmatrix} p_1 & 0 \\ 0 & p_2 \end{pmatrix},$$

where  $p_1 = \|Q_1 A^\top\|_{op} Q_1$ ,  $p_2 = \|Q_2 M^{-1}A\|_{op} Q_2$ . Finally, we define  $\varrho_t$  and  $z_t$  as follows,

$$\varrho_t = \omega_t - M^{-1}A\theta_t, \quad z_t = (\theta_t - \theta^*, \varrho_t - \varrho^*)^\top, \quad (28)$$

where  $\varrho^* = \omega^* - M^{-1}A\theta^*$ . Lyapunov function  $L(z_t)$  is:

$$L(z_t) = z_t^\top Q z_t. \quad (29)$$

**Motivation of Lyapunov Function** We consider the expected difference of iteration (22)-(23) as follows

$$\frac{1}{\alpha} \mathbb{E}[\theta_{t+1} - \theta_t | Y_{t-\tau}] = \mathbb{E}[-\hat{A}_t \omega_t | Y_{t-\tau}] \quad (30)$$

$$\frac{\alpha}{\beta} \frac{\mathbb{E}[\omega_{t+1} - \omega_t | Y_{t-\tau}]}{\alpha} = \mathbb{E}[\hat{A}_t \theta_t + \hat{b}_t - \hat{M}_t \omega_t | Y_{t-\tau}], \quad (31)$$

where  $Y_{t-\tau} = \{\theta_{t-\tau}, \omega_{t-\tau}, X_{t-\tau}\}$ , and the sequence  $X_t = \{S_0, A_0, S_1, A_1, \dots, S_t, A_t\}$  denotes a Markov chain according to Algorithm 1. The expectation is conditioned sufficiently in the past information of the underlying Markov chain. Approximating the left parts of (30)-(31) by derivatives, then we have the ordinary differential equation (ODE):

$$\begin{cases} \dot{\theta}(t) = -A\omega(t), \\ \frac{\alpha}{\beta} \dot{\omega}(t) = A\theta(t) + b - M\omega(t), \end{cases} \quad (32)$$

where  $\theta(t), \omega(t) \in \mathbb{R}^p$  of (32) are the functions that are defined on the continuous time  $(0, \infty)$ . The update rule of (22)-(23) can be thought of as a discretization of the ODE (32) that is known as *singular perturbation* ODE (chapter 11 of (Khalil and Grizzle 2002)), our goal is to provide a non-asymptotic analysis of  $\text{GES}(\lambda)$  according to the asymptotically stable equilibrium of ODE (32). According to Khalil and Grizzle (2002), the following Lyapunov function  $L(t)$  is widely used as a stability criteria for the ODE (32),

$$L(a, t) = a(\omega(t) - M^{-1}A\theta(t))^\top Q_2(\omega(t) - M^{-1}A\theta(t)) + (1-a)\theta^\top(t)Q_1\theta(t), \quad (33)$$

where  $a \in (0, 1)$ . Our  $L(z_t)$  (29) can be seen as a discretization of  $L(t)$  (33) after a proper choice of  $a$ , which inspires us to conduct the Lyapunov function  $L(z_t)$ .

**Lemma 1.** Under Assumption 1-3, there exists a positive scalar  $\tau$  such that:  $t \geq \tau$ ,

$$\begin{aligned} \mathbb{E}[L(z_{t+1})] - \mathbb{E}[L(z_t)] &\leq -\alpha \left(\frac{1}{2} \varkappa_1 - \frac{\alpha}{\beta} \varkappa_2\right) \mathbb{E}[L(z_t)] \\ &\quad + 2\alpha^2 \zeta^2 \lambda_{\max}(Q) \tilde{c}_b^2, \end{aligned} \quad (34)$$

where the constants  $\varkappa_1, \varkappa_2, \zeta =: C_1 + \frac{\beta}{\alpha} C_2, \tilde{c}_b$  are defined in Appendix D.

Finally, we know

$$\mathbb{E}[\|z_t\|_2^2] \leq (\lambda_{\min}(Q))^{-1} \mathbb{E}[L(z_t)],$$

applying the result of (34) recurrently, we have Theorem 3.

**Theorem 3.** Let  $\eta_1 = 4\zeta^2\tau^2(\|z_0\|_2 + \tilde{c}_b)^2 + \|z_0\|_2^2$ ,  $\eta_2 = \frac{2\kappa(Q)\zeta^2\lambda_{\max}(Q)\tilde{c}_b^2}{\frac{1}{2}\varkappa_1 - \frac{\alpha}{\beta}\varkappa_2}$ . Under Assumption 1-3, there exists a positive scalar  $\tau$  such that:  $t \geq \tau$ ,

$$\mathbb{E}[\|z_t\|_2^2] \leq \alpha^2 \eta_1 \left(1 - \frac{\alpha}{\lambda_{\max}(Q)} \left(\frac{1}{2} \varkappa_1 - \frac{\alpha}{\beta} \varkappa_2\right)\right)^{t-\tau} + \alpha \eta_2.$$

**Remark 3.** Recall  $z_t = (\theta_t - \theta^*, \varrho_t - \varrho^*)^\top$ , then  $\mathbb{E}[\|\theta_t - \theta^*\|_2^2] \leq \mathbb{E}[\|z_t\|_2^2]$ , which implies the expected mean square error of  $\theta_t - \theta^*$  is also upper-bounded by the result of Theorem 3. Furthermore, after a total computational cost of

$$\tau + \mathcal{O}\left(\frac{1}{\delta} \log \frac{1}{\delta}\right),$$

the Algorithm 1 outputs  $\theta_t$  closes to the optimal solution  $\theta^*$  as follows,

$$\mathbb{E}[\|\theta_t - \theta^*\|_2^2] \leq \mathcal{O}(\tau\delta).$$

**Remark 4.** Theorem 3 shows that the upper-bounded error consists of two different parts: the first error bound depends on both step-size and the size of samples, and this error decays geometrically as the number of iteration increases; while the second part is only determined by the step-sizes and it is independent of the number of iterations.

Algorithm	Step-size	Convergence Rate	TD Fixed Point
TD(0) (Nathaniel et al. 2015)	$\alpha_t = \mathcal{O}(t^{-\eta})$ $\eta \in (0, 1)$	$\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$	$\Phi^\top \Xi(\gamma P^\mu - I)\Phi\theta^* = -b$
TD(0) (Dalal et al. 2018a)	$\sum_{t=1}^{\infty} \alpha_t = \infty$	$\mathcal{O}\left(\frac{1}{T^\eta}\right)$ $\eta \in (0, 1)$	$\Phi^\top \Xi(\gamma P^\mu - I)\Phi\theta^* = -b$
TD(0) (Lakshminarayanan et al. 2018)	Constant	$\mathcal{O}\left(\frac{1}{T}\right)$	$\Phi^\top \Xi(\gamma P^\mu - I)\Phi\theta^* = -b$
GTD(0) (Dalal et al. 2018b)	$\sum_{t=1}^{\infty} \alpha_t = \infty, \frac{\beta_t}{\alpha_t} \rightarrow 0$	$\mathcal{O}\left(\left(\frac{1}{T}\right)^{\frac{1-\kappa}{3}}\right)$ $\kappa \in (0, 1)$	$\Phi^\top \Xi(\gamma P^\pi - I)\Phi\theta^* = -b$
GTD(0)/GTD2/TDC (Dalal et al. 2020)	$\alpha_t = \frac{1}{t^{\eta_1}}, \beta_t = \frac{1}{t^{\eta_2}}$ $0 < \eta_2 < \eta_1 < 1$	$\mathcal{O}\left(\frac{1}{T^{\eta_1}}\right)$	$\Phi^\top \Xi(\gamma P^\pi - I)\Phi\theta^* = -b$
GTB( $\lambda$ ) (Touati et al. 2018)	$\alpha_t, \beta_t = \mathcal{O}\left(\frac{1}{t}\right)$	$\mathcal{O}\left(\frac{1}{T}\right)$	$\Phi^\top \Xi(I - \gamma \lambda P^\mu)^{-1}(\gamma P^\pi - I)\Phi\theta^* = -b$
SARSA (Zou et al. 2019)	$\alpha_t = \mathcal{O}\left(\frac{1}{t}\right)$	$\mathcal{O}\left(\frac{\log^3(T)}{T}\right)$	$\Phi^\top \Xi(\gamma P^\mu - I)\Phi\theta^* = -b$
TDC (Xu et al. 2019)	$\max\{\alpha_t \log\left(\frac{1}{\alpha_t}\right), \alpha_t\}$ $\leq \min\left\{\frac{\ \theta_0 - \theta^*\ _2}{2^{t-1}}, C\right\}$	Linear	$\Phi^\top \Xi(\gamma P^\pi - I)\Phi\theta^* = -b$
TDC (Kaledin et al. 2020)	$\alpha_t = \frac{1}{t^v}, \beta_t = \frac{1}{t}$ $v \in (0, 1)$	$\mathcal{O}\left(\frac{1}{T}\right)$	$\Phi^\top \Xi(\gamma P^\pi - I)\Phi\theta^* = -b$
GES( $\lambda$ ) Theorem 3	$\alpha = \text{Constant}$	$\mathcal{O}(\alpha)$	$\Phi^\top \Xi(I - \gamma \lambda P^\pi)^{-1}(\gamma P^\pi - I)\Phi\theta^* = -b$

Table 1: Comparison of GTD family algorithms over performance measurement  $\mathbb{E}\|\theta_T - \theta^*\|_2^2$ .

## Related Works

In this section, we review existing finite-time performance of GTD algorithms over  $\|\theta_T - \theta^*\|_2^2$ .

Although the asymptotic analysis of GTD family has been established in (Sutton et al. 2009a,b; Maei 2011), which holds only in the limit as the number of iterations increases to infinity, and we can not get the information of convergence rate from asymptotic results. This is the main reason why we focus on the finite-time performance over  $\|\theta_T - \theta^*\|_2^2$ . It is noteworthy that Liu et al. (2015) firstly introduce *primal-dual gap error* to measure the convergence of GTD algorithm, we provide the discussion of finite-time primal-dual gap error analysis in Appendix E.

Nathaniel et al. (2015) proves that TD(0) (Sutton 1988) converges at  $\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)$  with the step-size  $\alpha_t = \mathcal{O}\left(\frac{1}{t^\eta}\right)$ ,  $\eta \in (0, 1)$ . Later, Dalal et al. (2018a) further explore the property of TD(0), they prove the convergence rate of TD(0) achieves  $\mathcal{O}\left(e^{-\frac{\lambda}{2}T^{1-\eta}} + \frac{1}{T^\eta}\right)$ , but it never reaches  $\mathcal{O}\left(\frac{1}{T}\right)$ , where  $\eta \in (0, 1)$ ,  $\lambda$  is the minimum eigenvalue of the matrix  $A^\top + A$ . Lakshminarayanan et al., (2018) show TD(0) converges at  $\mathcal{O}\left(\frac{1}{T}\right)$  with a more relaxed step-size than the works of (Nathaniel et al. 2015; Dalal et al. 2018a), it only requires a constant step-size. Recently, Dalal et al. (2018b) proves GTD(0) family algorithm (Sutton et al. 2009b) converges at  $\mathcal{O}\left(\left(\frac{1}{T}\right)^{\frac{1-\kappa}{3}}\right)$ , but never reach  $\mathcal{O}\left(\frac{1}{T}\right)$ , where  $\kappa \in (0, 1)$ . A very similar convergence rate appears in (Dalal et al. 2020), which considers TDC and GTD2. Touati et al. (2018) propose GTB( $\lambda$ )/GReTrace( $\lambda$ ), they prove the convergence rate

of GTB( $\lambda$ )/GReTrace( $\lambda$ ) reaches  $\mathcal{O}\left(\frac{1}{T}\right)$ . Zou et al. (2019) show SARSA with linear function approximation converges at the rate of  $\mathcal{O}\left(\frac{\log^3(T)}{T}\right)$ . Recently, Kaledin et al. (2020) further develop two timescale stochastic approximation with Markovian noise, and they show that TDC converges at a rate of  $\mathcal{O}\left(\frac{1}{T}\right)$  if  $\alpha_t = \frac{1}{t^v}$ ,  $\beta_t = \frac{1}{t}$ ,  $v \in (0, 1)$ .

Theorem 3 illustrates our GES( $\lambda$ ) achieves the convergence rate at the same order with step-size. Although Xu et al., (2019) prove TDC converges at a linear convergence rate, they require a fussy blockwise diminishing step-size condition:  $\max\{\alpha_t \log\left(\frac{1}{\alpha_t}\right), \alpha_t\} \leq \left\{\min\left\{\frac{\|\theta_0 - \theta^*\|_2}{2^{t-1}}, C\right\}, \alpha_t = \mathcal{O}\left(\frac{1}{(t+1)^{\eta_1}}\right), \beta_t = \mathcal{O}\left(\frac{1}{(t+1)^{\eta_2}}\right), 0 < \eta_2 < \eta_1 < 1\right\}$ , where  $C$  is a constant. Apparently, our Theorem 2 requires a simpler condition of step-size than Xu et al., (2019). It is noteworthy that our Theorem 2 does not require an additional projection step (that is unnecessary in practice) that appears in (Xu et al. 2019).

Significantly, the finite-time performances of (Dalal et al. 2018a; Lakshminarayanan et al. 2018) requires an additional assumption that all the samples required to update the function parameters are i.i.d. In this paper, we remove this condition and achieve a better result than theirs.

## Experiments

In this section, we test the capacity of GES( $\lambda$ ) for off-policy evaluation in three typical domains: MountainCar, Baird Star (Baird 1995), Two-State MDP (Touati et al. 2018). We

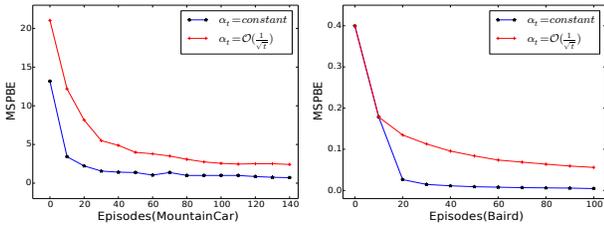


Figure 2: Comparison between a constant step-size and  $\frac{1}{\sqrt{t}}$ .

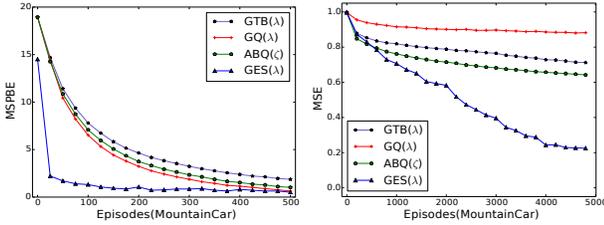


Figure 3: MSPBE and MSE comparison on MountainCar.

compare GES( $\lambda$ ) with three state-of-art algorithms: GQ( $\lambda$ ) (Maei and Sutton 2010), ABQ( $\zeta$ ) (Mahmood et al. 2017b), GTB( $\lambda$ ) (Touati et al. 2018) over two typical measurements: MSPBE and mean square error (MSE). We choose those three algorithms as baselines since they are all learning via expected TD-error  $\delta_t^{\text{ES}}$ , which is same as GES( $\lambda$ ).

**Feature Map and Parameters** Recall the states and actions of MountainCar:  $\mathcal{S} = \{(\text{Velocity}, \text{Position})\} = [-0.07, 0.07] \times [-1.2, 0.6]$ ,  $\mathcal{A} = \{\text{left}, \text{neutral}, \text{right}\}$ . In this example, if  $\text{Velocity} > 0$ , we use behavior policy  $\mu = (\frac{1}{100}, \frac{1}{100}, \frac{98}{100})$ ,  $\pi = (\frac{1}{10}, \frac{1}{10}, \frac{8}{10})$ ; else  $\mu = (\frac{98}{100}, \frac{1}{100}, \frac{1}{100})$ ,  $\pi = (\frac{8}{10}, \frac{1}{10}, \frac{1}{10})$ . Since the state space is continuous, we use an open *tile coding*<sup>3</sup> software to extract feature of states. We set the number of tilings to be 4, and there are no white noise features. The performance is an average 5 runs, and each run contains 5000 episodes. As suggested by Sutton and Barto (2018), we set all the initial parameters to be 0, which is optimistic about causing extensive exploration.

The Baird Star is an episodic seven states MDP with two actions: dashed action and solid action. In this example, we set the behavior policy  $\mu(\cdot|\text{dashed}) = \frac{6}{7}$ ,  $\mu(\cdot|\text{solid}) = \frac{1}{7}$  and target policy  $\pi(\cdot|\text{solid}) = 1$ . We choose the feature map matrix as follows  $\Phi = \begin{pmatrix} 2I_{7 \times 7} & \mathbf{1}_{7 \times 1} & \mathbf{0}_{7 \times 8} \\ \mathbf{0}_{7 \times 8} & 2I_{7 \times 7} & \mathbf{1}_{7 \times 1} \end{pmatrix}$ , where  $\mathbf{0}$  denotes a matrix whose elements are all 0, and  $\mathbf{1}_{7 \times 1}$  denotes a vector whose elements are all 1. The dynamics of Two-State MDP is presented in Example 1. We set  $\lambda = 0.99$ ,  $\gamma = 0.99$  in all the experiments. The MSPBE/MSE distribution is computed over the combination of step-size,  $(\alpha_t, \frac{\beta_t}{\alpha_t}) \in [0.1 \times 2^j | j = -10, -9, \dots, -1, 0]^2$ .

**Effect of Step-size** Figure 2 shows the comparison of the

<sup>3</sup><http://incompleteideas.net/rlai.cs.ualberta.ca/RLAI/RLtoolkit/tilecoding.html>

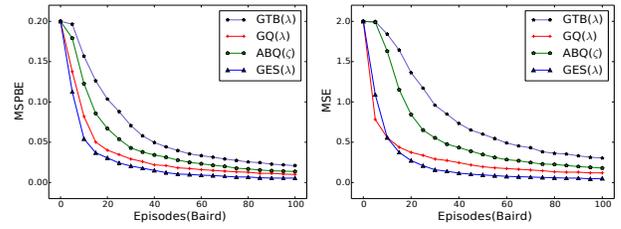


Figure 4: MSPBE and MSE comparison on Baird Star.

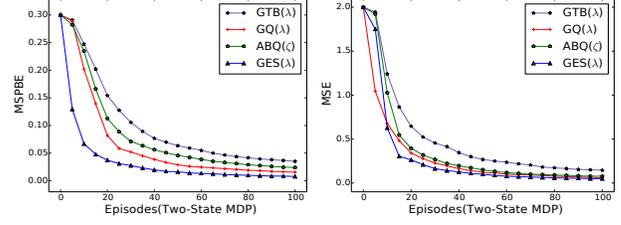


Figure 5: MSPBE and MSE comparison on Two-State MDP.

empirical MSPBE performance between a constant step-size and the decay step-size  $\mathcal{O}(\frac{1}{\sqrt{t}})$ . Result of Figure 2 illustrates that the GES( $\lambda$ ) with a proper constant step-size converges significantly faster than the learning with step-size  $\mathcal{O}(\frac{1}{\sqrt{t}})$ , which also supports Theorem 2: learning with a proper constant step-size can reach a very faster rate.

**Comparison of Empirical MSPBE and MSE** In this section, we use empirical MSPBE  $= \frac{1}{2} \|\hat{b} + \hat{A}\theta\|_{\hat{M}^{-1}}^2$  to evaluate the performance, where we evaluate  $\hat{A}$ ,  $\hat{b}$ , and  $\hat{M}$  according to their unbiased estimators by Monte Carlo method with 5000 episodes, and our implementation of MSPBE is inspired by (Touati et al. 2018). Besides, we also compare the performance over a common measurement empirical MSE:  $\text{MSE} = \|\Phi\theta - q^\pi\|_{\Xi}^2 / \|q^\pi\|_{\Xi}^2$ , where  $q^\pi$  is estimated by simulating the target policy  $\pi$  and averaging the discounted cumulative rewards over trajectories. The combination of step-size for MSE is the same as previous empirical MSPBE.

Results in Figure 3 to 5 show that our GES( $\lambda$ ) learns faster with a better performance than GQ( $\lambda$ ), ABQ( $\zeta$ ) and GTB( $\lambda$ ). Besides, GES( $\lambda$ ) converges with a lower variance. In the Two-State MDP and Baird Star experiments, GES( $\lambda$ ) outperforms the baselines slightly. This is because both Two-State MDP and Baird Star are relatively easy; many gradient TD learning could learn a convergent result. While the advantage of GES( $\lambda$ ) over baselines becomes more significant in the MountainCar domain, which shows that GES( $\lambda$ ) is more robust than baselines in the more difficult task.

## Conclusion

We propose GES( $\lambda$ ) that extends Expected Sarsa( $\lambda$ ) with linear function approximation. We prove GES( $\lambda$ ) learns the optimal solution at a linear convergence rate, which is comparable to extensive GTD algorithms. The primal-dual gap

error of  $\text{GES}(\lambda)$  matches the best-known theoretical results, but we require a simpler condition of step-size. Finally, we conduct experiments to verify the effectiveness of  $\text{GES}(\lambda)$ .

### Acknowledgements

This work was supported by Natural Science Foundation of China (No. 61925603) Zhejiang Lab (No. 2019KC0AD02). The authors would like to thank Ahmed Touati for providing a number of valuable comments for us to polish the paper.

### References

- Baird, L. 1995. Residual algorithms: Reinforcement learning with function approximation. In *ICML*, 30–37.
- Bertsekas, D. P. 2011. Temporal difference methods for general projected equations. *IEEE Transactions on Automatic Control* 56(9): 2128–2139.
- Bertsekas, D. P. 2012. *Dynamic Programming and Optimal Control*, volume 2. Athena scientific Belmont, MA.
- Bhandari, J.; Russo, D.; Singal, R.; et al. 2018. A finite time analysis of temporal difference learning with linear function approximation. *COLT*.
- Dalal, G.; Szorenyi, B.; Thoppe, G.; and Mannor, S. 2018a. Finite sample analyses for  $\text{td}(0)$  with function approximation. In *AAAI2018*.
- Dalal, G.; Szorenyi, B.; Thoppe, G.; and Mannor, S. 2018b. Finite Sample Analysis of Two-Timescale Stochastic Approximation with Applications to Reinforcement Learning. In *COLT*.
- Dalal, G.; Szorenyi, B.; Thoppe, G.; et al. 2020. A Tale of Two-Timescale Reinforcement Learning with the Tightest Finite-Time Bound. *AAAI*.
- Dann, C.; Neumann, G.; Peters, J.; et al. 2014. Policy evaluation with temporal differences: A survey and comparison. *JMLR* 15(1): 809–883.
- Du, S. S.; Chen, J.; Li, L.; Xiao, L.; and Zhou, D. 2017. Stochastic variance reduction methods for policy evaluation. *ICML*.
- Du, S. S.; and Hu, W. 2019. Linear convergence of the primal-dual gradient method for convex-concave saddle point problems without strong convexity. *AISTATS*.
- Geist, M.; and Scherrer, B. 2014. Off-policy learning with eligibility traces: A survey. *JMLR*.
- Ghosh, D.; and Bellemare, M. G. 2020. Representations for Stable Off-Policy Reinforcement Learning. *ICML*.
- Gupta, H.; Srikant, R.; Ying, L.; et al. 2019. Finite-time performance bounds and adaptive learning rate selection for two time-scale reinforcement learning. In *NeurIPS*, 4704–4713.
- Kakade, S.; Shalev-Shwartz, S.; Tewari, A.; et al. 2009. On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization.
- Kaledin, M.; Moulines, E.; Naumov, A.; Tadic, V.; and Wai, H.-T. 2020. Finite time analysis of linear two-timescale stochastic approximation with Markovian noise. *COLT*.
- Khalil, H. K.; and Grizzle, J. W. 2002. *Nonlinear systems*, volume 3. Prentice hall Upper Saddle River, NJ.
- Lagoudakis, M. G.; and Parr, R. 2003. Least-squares policy iteration. *JMLR* 4: 1107–1149.
- Lakshminarayanan, C.; Szepesvari, Csaba; et al. 2018. Linear Stochastic Approximation: How Far Does Constant Step-Size and Iterate Averaging Go? In *AISTATS*.
- Liu, B.; Liu, J.; Ghavamzadeh, M.; Mahadevan, S.; and Petrik, M. 2015. Finite-Sample Analysis of Proximal Gradient TD Algorithms. In *UAI*.
- Maei, H. R. 2011. *Gradient temporal-difference learning algorithms*. Ph.D. thesis, University of Alberta Edmonton, Alberta.
- Maei, H. R.; and Sutton, R. S. 2010.  $\text{GQ}(\lambda)$ : A general gradient algorithm for temporal-difference prediction learning with eligibility traces. In *Proceedings of the third conference on artificial general intelligence*, volume 1, 91–96.
- Mahmood, A. R.; Yu, H.; Sutton, R. S.; et al. 2017b. Multi-step off-policy learning without importance sampling ratios.
- Nathaniel, K.; Prashanth, L.; Prashanth, L.; et al. 2015. On TD(0) with function approximation: Concentration bounds and a centered variant with exponential convergence. In *ICML*.
- Nemirovski, A.; Juditsky, A.; Lan, G.; and Shapiro, A. 2009. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization* 19(4): 1574–1609.
- Precup, D.; Sutton, R. S.; and Singh, S. P. 2000. Eligibility Traces for Off-Policy Policy Evaluation. In *International Conference on Machine Learning*, 759–766. Citeseer.
- Rockafellar, R. T. 1970. *Convex analysis*. 28. Princeton university press.
- Rubinstein, R. Y.; and Kroese, D. P. 2016. *Simulation and the Monte Carlo method*, volume 10. John Wiley & Sons.
- Srikant; and Ying, L. 2019. Finite-Time Error Bounds For Linear Stochastic Approximation and TD Learning. In *COLT*.
- Sutton, R. S. 1988. Learning to predict by the methods of temporal differences. *Machine learning* 3(1): 9–44.
- Sutton, R. S.; and Barto, A. G. 1998. *Reinforcement learning: An introduction*. MIT press Cambridge.
- Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.
- Sutton, R. S.; Maei, H. R.; Precup, D.; Bhatnagar, S.; Silver, D.; Szepesvári, C.; and Wiewiora, E. 2009a. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *ICML*.
- Sutton, R. S.; Maei, H. R.; Szepesvári, C.; et al. 2009b. A Convergent  $O(n)$  Temporal-difference Algorithm for Off-policy Learning with Linear Function Approximation. In *NeurIPS*.

- Sutton, R. S.; Mahmood, A. R.; White, M.; et al. 2016. An emphatic approach to the problem of off-policy temporal-difference learning. *JMLR* 17(1): 2603–2631.
- Thomas, P. S. 2015. *Safe reinforcement learning*. Ph.D. thesis, University of Massachusetts Libraries.
- Touati, A.; Bacon, P.-L.; Precup, D.; and Vincent, P. 2018. Convergent Tree-Backup and Retrace with Function Approximation. In *ICML*.
- Voloshin, C.; Le, H. M.; Jiang, N.; and Yue, Y. 2019. Empirical Study of Off-Policy Policy Evaluation for Reinforcement Learning. *arXiv preprint arXiv:1911.06854* .
- Wang, G.; Li, B.; Giannakis, G. B.; et al. 2019. A multistep Lyapunov approach for finite-time analysis of biased stochastic approximation. *arXiv preprint arXiv:1909.04299* .
- Wang, Y.; Chen, W.; Liu, Y.; Ma, Z.-M.; and Liu, T.-Y. 2017. Finite Sample Analysis of the GTD Policy Evaluation Algorithms in Markov Setting. In *Advances in Neural Information Processing Systems(NeurIPS)*.
- Xu, T.; Zou, S.; Liang, Y.; et al. 2019. Two time-scale off-policy TD learning: Non-asymptotic analysis over Markovian samples. In *NeurIPS*.
- Zou, S.; Xu, T.; Liang, Y.; et al. 2019. Finite-sample analysis for SARSA with linear function approximation. In *NeurIPS*.