mathematics

MDPI

Article

On Convergence Rate of MRetrace

Xingguo Chen [1], Wangrong Qin [1], Yu Gong [1], Shangdong Yang [1] and Wenhao Wang [2,3,*]

[1] Jiangsu Key Laboratory of Big Data Security & Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing 210023, China; chenxg@njupt.edu.cn (X.C.); sdyang@njupt.edu.cn (S.Y.)

[2] College of Electronic Engineering, National University of Defense Technology, Changsha 410073, China

[3] Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China

* Correspondence: wangwenhao11@nudt.edu.cn

Abstract: Off-policy is a key setting for reinforcement learning algorithms. In recent years, the stability of off-policy learning for value-based reinforcement learning has been guaranteed even when combined with linear function approximation and bootstrapping. Convergence rate analysis is currently a hot topic. However, the convergence rates of learning algorithms vary, and analyzing the reasons behind this remains an open problem. In this paper, we propose an essentially simplified version of a convergence rate to generate general off-policy temporal difference learning algorithms. We emphasize that the primary determinant influencing convergence rate is the minimum eigenvalue of the key matrix. Furthermore, we conduct a comparative analysis of the influencing factor across various off-policy learning algorithms in diverse numerical scenarios. The experimental findings validate the proposed determinant, which serves as a benchmark for the design of more efficient learning algorithms.

Keywords: finite sample analysis; off-policy learning; minimum eigenvalues; MRetrace

MSC: 68T05

check for updates

Citation: Chen, X.; Qin, W.; Gong, Y.; Yang, S; Wang, W. On Convergence Rate of MRetrace. *Mathematics* **2024**, *12*, 2930. https://doi.org/10.3390/math12182930

Academic Editor: Giovanni Stabile

Received: 2 August 2024

Revised: 15 September 2024

Accepted: 16 September 2024

Published: 20 September 2024

1. Introduction

Off-policy learning generates experience data by a behavior policy and learns a different target policy. Off-policy TD learning with a linear function approximation may diverge in counterexamples known as “the deadly triad” [1]. The fundamental reason is that the key matrix of off-policy TD is not guaranteed to be positive definite [2]. In the recent 30 years, the main research focused on the convergence guarantee of off-policy algorithms via the construction of positive definite matrices, e.g., Bellman residual (BR) [3], gradient temporal difference (GTD) [4], fast gradient temporal difference (GTD2) and TD with gradient correction (TDC) [5], emphatic TD (ETD) [2], and modified Retrace (MRetrace) [6].

Recently, due to the guarantee of convergence, more research has paid attention to the convergence rate analysis of reinforcement learning algorithms. Dalal et al. [7] proposed convergence rates both in expectation and with a high probability for one-timescale temporal difference learning algorithms. Dalal et al. [8], Gupta et al. [9], Xu et al. [10], and Dalal et al. [11] obtained convergence rates with a high probability for two-timescale temporal difference learning algorithms. Durmus et al. [12] proposed tight high-probability bounds for linear stochastic approximation with a fixed step size. For control settings, Xu and Liang [13] proposed convergence rates for Greedy-GQ. Zhang et al. [14] proposed convergence rates for projected SARSA. Wang et al. [15] proposed convergence rates with a high probability for distributionally robust Q-learning.

However, the above analysis did not answer the following questions: Which of these algorithms is faster? Which one should we choose? The purpose of this paper is to give an intuitive comparison of the convergence rate.

Our contributions: (1) We propose a simplified version of the expected convergence rate theorem. (2) We analyze the core elements of the convergence rates by assuming the same settings for each algorithm, and we find that the main factor affecting convergence rate is the minimum eigenvalue of the key matrix. (3) We calculate core elements of different temporal difference learning algorithms for several environmental examples and validate by experimental studies.

2. Background

This section introduces MDP and reinforcement learning algorithms with their key matrices.

2.1. Markov Decision Process

Consider a discounted Markov decision process $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \gamma \rangle$, where $\mathcal{S}$ is a state space, $|\mathcal{S}| = n$, $\mathcal{A}$ is an action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$ is a transition function, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is a reward function, and $\gamma \in [0,1)$ is a discount factor. The state value function is

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s\right], \tag{1}$$

where r_t is an immediate reward, and π is a policy used to select action a in state s with a probability $\pi(a|s)$. The value function is approximated with a linear function, as follows:

$$V(s) \approx V_{\theta}(s) = \theta^{\top}\phi(s) = \sum_{i=1}^{m} \theta_i \phi_i(s), \tag{2}$$

where θ is the weight vector, and $\phi(s)$ is the feature vector of state s. This paper is concerned with off-policy learning, where a different behavior policy μ generates experiences $\langle s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1} \rangle$.

2.2. Learning Algorithms and Their Key Matrices

Learning algorithms and their key matrices are summarized in Table 1.

Table 1. The general solution expressions for each algorithm ($\theta = \mathbf{A}^{-1}\mathbf{b}$).

Algorithm	Key Matrix A	Positive Definite	b
Off-policy TD	$\mathbf{A}_{\text{off}} = \Phi^{\top}\mathbf{D}_{\mu}(\mathbf{I} - \gamma\mathbf{P}_{\pi})\Phi$	×	$\mathbf{b}_{\text{off}} = \Phi^{\top}\mathbf{D}_{\mu}r_{\pi}$
Retrace	$\Phi^{\top}\mathbf{D}_{\mu}\mathbf{D}_{c}(\mathbf{I} - \gamma\mathbf{P}_{\pi})\Phi$	×	$\Phi^{\top}\mathbf{D}_{\mu}r_{c}$
BR	$\Phi^{\top}(\mathbf{I} - \gamma\mathbf{P}_{\pi})^{\top}\mathbf{D}_{\mu}(\mathbf{I} - \gamma\mathbf{P}_{\pi})\Phi$	✓	$\Phi^{\top}(\mathbf{I} - \gamma\mathbf{P}_{\pi})^{\top}\mathbf{D}_{\mu}r_{\pi}$
GTD	$\begin{pmatrix} \sqrt{\eta}\mathbf{I} & \mathbf{A}_{\text{off}} \\ -\mathbf{A}_{\text{off}}^{\top} & 0 \end{pmatrix}$	✓	$\begin{pmatrix} \mathbf{b}_{\text{off}} \\ 0 \end{pmatrix}$
GTD2	$\begin{pmatrix} \sqrt{\eta}\mathbf{C} & \mathbf{A}_{\text{off}} \\ -\mathbf{A}_{\text{off}}^{\top} & 0 \end{pmatrix}$	✓	$\begin{pmatrix} \mathbf{b}_{\text{off}} \\ 0 \end{pmatrix}$
TDC	$\mathbf{A}_{\text{off}}^{\top}\mathbf{C}^{-1}\mathbf{A}_{\text{off}}$	✓	$\mathbf{A}_{\text{off}}^{\top}\mathbf{C}^{-1}\mathbf{b}_{\text{off}}$
ETD	$\Phi^{\top}\mathbf{D}_{f}(\mathbf{I} - \gamma\mathbf{P}_{\pi})\Phi$	✓	$\Phi^{\top}\mathbf{D}_{f}r_{\pi}$
MRetrace	$\Phi^{\top}\mathbf{D}_{\mu}(\mathbf{I} - \gamma\mathbf{D}_{x}\mathbf{P}_{\pi})\Phi$	✓	$\mathbf{b}_{\text{off}}$

2.2.1. Off-Policy TD

The update rule of off-policy TD [2] is as follows:

$$\begin{aligned} \theta_{t+1} &\doteq \theta_t + \rho_t\alpha_t(r_{t+1} + \gamma\theta_t^{\top}\phi_{t+1} - \theta_t^{\top}\phi_t)\phi_t \\ &= \theta_t + \alpha_t\Big(\rho_t r_{t+1}\phi_t - \rho_t\phi_t(\phi_t - \gamma\phi_{t+1})^{\top}\theta_t\Big) \\ &= \theta_t + \alpha_t\big(\mathbf{b}_{\text{off},t} - \mathbf{A}_{\text{off},t}\theta_t\big), \end{aligned} \tag{3}$$

Its key matrix is

$$\begin{aligned}
\mathbf{A}_{\text{off}} &= \lim_{t\to\infty} \mathbb{E}[\mathbf{A}_{\text{off},t}] = \lim_{t\to\infty} \mathbb{E}_{\mu}\left[\rho_t\phi_t(\phi_t - \gamma\phi_{t+1})^\top\right] \\
&= \sum_s d_\mu(s)\mathbb{E}_\mu\left[\rho_t\phi_t(\phi_t - \gamma\phi_{t+1})^\top | S_t = s\right] \\
&= \sum_s d_\mu(s)\sum_a \mu(a|s)\sum_{s'} T(s,a,s')\frac{\pi(a|s)}{\mu(a|s)}\phi(s)\big(\phi(s) - \gamma\phi(s')\big)^\top \\
&= \sum_s d_\mu(s)\phi(s)\big(\phi(s) - \gamma\sum_a \pi(a|s)\sum_{s'} T(s,a,s')\phi(s')\big)^\top \\
&= \sum_s d_\mu(s)\phi(s)\big(\phi(s) - \gamma\sum_{s'}[\mathbf{P}_\pi]_{ss'}\phi(s')\big)^\top \\
&= \mathbf{\Phi}^\top\mathbf{D}_\mu(\mathbf{I} - \gamma\mathbf{P}_\pi)\mathbf{\Phi},
\end{aligned} \tag{4}$$

$$\begin{aligned}
\mathbf{b}_{\text{off}} &= \lim_{t\to\infty} \mathbb{E}[\mathbf{b}_{\text{off},t}] = \lim_{t\to\infty} \mathbb{E}_\mu[\rho_t r_{t+1}\phi_t] \\
&= \sum_s d_\mu(s)\mathbb{E}_\mu[\rho_t r_{t+1}\phi_t | S_t = s] \\
&= \sum_s d_\mu(s)\sum_a \mu(a|s)\frac{\pi(a|s)}{\mu(a|s)} r_{t+1}\phi(s) \\
&= \sum_s d_\mu(s)\phi(s)\sum_a \pi(a|s)\sum_{s'} T(s,a,s')R(s,a,s') \\
&= \mathbf{\Phi}^\top\mathbf{D}_\mu r_\pi,
\end{aligned} \tag{5}$$

where r_π is an expected reward vector under policy π with each component being

$$r_\pi(s) = \sum_a\sum_{s'} \pi(a|s)R(s,a,s'). \tag{6}$$

2.2.2. Retrace(0)

The update rule of Retrace(0) [16] is as follows:

$$\begin{aligned}
\theta_{t+1} &\doteq \theta_t + c_t\alpha_t\Big(r_{t+1} + \gamma\theta_t^\top\mathbb{E}_\pi[\phi_{t+1}] - \theta_t^\top\phi_t\Big)\phi_t \\
&= \theta_t + \alpha_t\Big(c_t r_{t+1}\phi_t - c_t\phi_t(\phi_t - \gamma\mathbb{E}_\pi[\phi_{t+1}])^\top\theta_t\Big) \\
&= \theta_t + \alpha_t\big(\mathbf{b}_{\text{Retrace}(0),t} - \mathbf{A}_{\text{Retrace}(0),t}\theta_t\big),
\end{aligned} \tag{7}$$

where $c_t = \min(1,\rho_t)$, $\mathbb{E}_\pi[\phi_{t+1}] = \sum_a \pi(a|s_{t+1})\phi(s_{t+1})$. The key matrix of the expected Retrace's update (7) is

$$\begin{aligned}
\mathbf{A}_{\text{Retrace}(0)} &= \lim_{t\to\infty} \mathbb{E}[\mathbf{A}_{\text{Retrace}(0),t}] = \lim_{t\to\infty} \mathbb{E}_\mu\left[c_t\phi_t(\phi_t - \gamma\mathbb{E}_\pi[\phi_{t+1}])^\top\right] \\
&= \sum_s d_\mu(s)\mathbb{E}_\mu\left[c_t\phi(s)(\phi(s) - \gamma\mathbb{E}_\pi[\phi(s')])^\top\right] \\
&= \sum_s d_\mu(s)\phi(s)\sum_a \mu(a|s)\min\left(1,\frac{\pi(a|s)}{\mu(a|s)}\right)(\phi(s) - \gamma\mathbb{E}_\pi[\phi(s')])^\top \\
&= \sum_s d_\mu(s)\phi(s)\sum_a \min\left(\mu(a|s),\pi(a|s)\right)\left(\phi(s) - \gamma\sum_{s'}[\mathbf{P}_\pi]_{ss'}\phi(s')\right)^\top \\
&= \mathbf{\Phi}^\top\mathbf{D}_\mu\mathbf{D}_c(\mathbf{I} - \gamma\mathbf{P}_\pi)\mathbf{\Phi},
\end{aligned} \tag{8}$$

where $\mathbf{D}_c$ is the $n \times n$ diagonal matrix with d_c on its diagonal, and each component of d_c is

$$d_c(s) = \sum_a \min(\mu(a|s),\pi(a|s)). \tag{9}$$

$$\begin{aligned}
\mathbf{b}_{\text{ Retrace(0)}} &= \lim_{t\to\infty}\mathbb{E}[\mathbf{b}_{\text{Retrace(0)},t}] = \lim_{t\to\infty}\mathbb{E}_\mu[c_t r_{t+1}\phi_t] \\
&= \sum_s d_\mu(s)\mathbb{E}_\mu[c_t r_{t+1}\phi_t|S_t = s] \\
&= \sum_s d_\mu(s)\sum_a \mu(a|s)\min\left(1, \frac{\pi(a|s)}{\mu(a|s)}\right) r_{t+1}\phi(s) \\
&= \sum_s d_\mu(s)\phi(s)\sum_a \min\left(\mu(a|s), \pi(a|s)\right)\sum_{s'} T(s,a,s')R(s,a,s') \\
&= \Phi^\top \mathbf{D}_\mu r_c,
\end{aligned} \tag{10}$$

where $r_c = \sum_a \min\left(\mu(a|s), \pi(a|s)\right)\sum_{s'} T(s,a,s')R(s,a,s')$.

2.2.3. Naive Bellman Residual

The update rule of the naive Bellman residual [3] is as follows:

$$\begin{aligned}
\theta_{t+1} \doteq& \theta_t + \rho_t\alpha_t(r_{t+1} + \gamma\theta_t^\top\phi_{t+1} - \theta_t^\top\phi_t)(\phi_t - \gamma\mathbb{E}_\pi[\phi_{t+1}]) \\
=& \theta_t + \alpha_t\Big(\rho_t r_{t+1}(\phi_t - \gamma\phi_{t+1}) - \rho_t(\phi_t - \gamma\phi_{t+1})(\phi_t - \gamma\mathbb{E}_\pi[\phi_{t+1}])^\top\theta_t\Big) \\
=& \theta_t + \alpha_t\big(\mathbf{b}_{\text{BR},t} - \mathbf{A}_{\text{BR},t}\theta_t\big),
\end{aligned} \tag{11}$$

Its key matrix is

$$\begin{aligned}
&\mathbf{A}_{\text{BR}} = \lim_{t\to\infty}\mathbb{E}[\mathbf{A}_{\text{BR},t}] = \lim_{t\to\infty}\mathbb{E}_\mu\Big[\rho_t(\phi_t - \gamma\mathbb{E}_\pi[\phi_{t+1}])(\phi_t - \gamma\phi_{t+1})^\top\Big] \\
&= \sum_s d_\mu(s)\mathbb{E}_\mu\Big[\rho_t(\phi_t - \gamma\mathbb{E}_\pi[\phi_{t+1}])(\phi_t - \gamma\phi_{t+1})^\top|S_t = s\Big] \\
&= \sum_s d_\mu(s)\sum_a \mu(a|s)\sum_{s'} T(s,a,s')\frac{\pi(a|s)}{\mu(a|s)}(\phi(s) - \gamma\mathbb{E}_\pi[\phi(s')])\big(\phi(s) - \gamma\phi(s')\big)^\top \\
&= \sum_s d_\mu(s)\sum_a\sum_{s'} T(s,a,s')\pi(a|s)(\phi(s) - \gamma\mathbb{E}_\pi[\phi(s')])\big(\phi(s) - \gamma\phi(s')\big)^\top \\
&= \sum_s d_\mu(s)(\phi(s) - \gamma\mathbb{E}_\pi[\phi(s')])\big(\phi(s) - \gamma\sum_a \pi(a|s)\sum_{s'} T(s,a,s')\phi(s')\big)^\top \\
&= \sum_s d_\mu(s)(\phi(s) - \gamma\mathbb{E}_\pi[\phi(s')])\big(\phi(s) - \gamma\sum_{s'}[\mathbf{P}_\pi]_{ss'}\phi(s')\big)^\top \\
&= \Phi^\top(\mathbf{I} - \gamma\mathbf{P}_\pi)^\top\mathbf{D}_\mu(\mathbf{I} - \gamma\mathbf{P}_\pi)\Phi,
\end{aligned} \tag{12}$$

$$\begin{aligned}
\mathbf{b}_{\text{BR}} &= \lim_{t\to\infty}\mathbb{E}[\mathbf{b}_{\text{BR},t}] = \lim_{t\to\infty}\mathbb{E}_\mu[\rho_t r_{t+1}(\phi_t - \gamma\mathbb{E}_\pi[\phi_{t+1}])] \\
&= \sum_s d_\mu(s)\mathbb{E}_\mu[\rho_t r_{t+1}(\phi_t - \gamma\mathbb{E}_\pi[\phi_{t+1}])|S_t = s] \\
&= \sum_s d_\mu(s)\sum_a \mu(a|s)\frac{\pi(a|s)}{\mu(a|s)} r_{t+1}(\phi(s) - \gamma\mathbb{E}_\pi[\phi(s')])) \\
&= \sum_s d_\mu(s)(\phi(s) - \gamma\mathbb{E}_\pi[\phi(s')]))\sum_a \pi(a|s)\sum_{s'} T(s,a,s')R(s,a,s') \\
&= \Phi^\top(\mathbf{I} - \gamma\mathbf{P}_\pi)^\top\mathbf{D}_\mu r_\pi,
\end{aligned} \tag{13}$$

2.2.4. GTD

The update rule of the GTD [4] algorithm is as follows:

$$\begin{aligned}
\omega_{t+1} &= \omega_t + \beta_t(\delta_t\phi_t - \omega_t), \\
\theta_{t+1} &= \theta_t + \alpha_t(\phi_t - \gamma\phi_t')\phi_t^\top\omega_t,
\end{aligned} \tag{14}$$

where the TD error $\delta_t = r_t + (\gamma\phi_t' - \phi_t)^\top\theta_t$.

Let $g_t = (\omega_t^\top/\sqrt{\eta}, \theta_t^\top)^\top$; thus, we can obtain

$$g_{t+1} = g_t + \alpha_t\sqrt{\eta}(\mathbf{b}_{GTD,t+1} - \mathbf{A}_{GTD,t+1}g_t), \tag{15}$$

where $\mathbf{b}_{GTD,t+1}^\top = \left(r_t\phi_t^\top, 0^\top\right)$ and

$$\mathbf{A}_{GTD,t+1} = \begin{pmatrix} \sqrt{\eta}\mathbf{I} & \phi_t(\phi_t - \gamma\phi_t')^\top \\ (\gamma\phi_t' - \phi_t)\phi_t^\top & 0 \end{pmatrix}. \tag{16}$$

Its key matrix is

$$\mathbf{A}_{\text{GTD}} = \lim_{t\to\infty}\mathbb{E}[\mathbf{A}_{\text{GTD},t}] = \begin{pmatrix} \sqrt{\eta}\mathbf{I} & \mathbf{A}_{\text{off}} \\ -\mathbf{A}_{\text{off}}^\top & 0 \end{pmatrix}, \tag{17}$$

$$\mathbf{b}_{\text{GTD}} = \lim_{t\to\infty}\mathbb{E}[\mathbf{b}_{\text{GTD},t}] = \begin{pmatrix} \mathbf{\Phi}^\top\mathbf{D}_\mu r_\pi \\ 0 \end{pmatrix}, \tag{18}$$

2.2.5. GTD2

The update rule of the GTD2 [5] algorithm is as follows:

$$\begin{aligned} \omega_{t+1} &= \omega_t + \beta_t(\delta_t - \phi_t^\top\omega_t)\phi_t, \\ \theta_{t+1} &= \theta_t + \alpha_t(\phi_t - \gamma\phi_t')\phi_t^\top\omega_t, \end{aligned} \tag{19}$$

where the TD error $\delta_t = r_t + (\gamma\phi_t' - \phi_t)^\top\theta_t$.

Let $g_t = (\omega_t^\top/\sqrt{\eta}, \theta_t^\top)^\top$; thus, we can obtain

$$g_{t+1} = g_t + \alpha_t\sqrt{\eta}(\mathbf{b}_{GTD2,t+1} - \mathbf{A}_{GTD2,t+1}g_t), \tag{20}$$

where $\mathbf{b}_{GTD2,t+1}^\top = \left(r_t\phi_t^\top, 0^\top\right)$ and

$$\mathbf{A}_{GTD2,t+1} = \begin{pmatrix} \sqrt{\eta}\phi_t\phi_t^\top & \phi_t(\phi_t - \gamma\phi_t')^\top \\ (\gamma\phi_t' - \phi_t)\phi_t^\top & 0 \end{pmatrix}.$$

Its key matrix is

$$\mathbf{A}_{\text{GTD2}} = \lim_{t\to\infty}\mathbb{E}[\mathbf{A}_{\text{GTD2},t}] = \begin{pmatrix} \sqrt{\eta}\mathbf{C} & \mathbf{A}_{\text{off}} \\ -\mathbf{A}_{\text{off}}^\top & 0 \end{pmatrix}, \tag{21}$$

$$\mathbf{b}_{\text{GTD2}} = \lim_{t\to\infty}\mathbb{E}[\mathbf{b}_{\text{GTD2},t}] = \begin{pmatrix} \mathbf{\Phi}^\top\mathbf{D}_\mu r_\pi \\ 0 \end{pmatrix}, \tag{22}$$

where $\mathbf{C} = \mathbb{E}[\phi\phi^\top]$.

2.2.6. TDC

The update rule of the TDC [5] algorithm is as follows:

$$\begin{aligned} \omega_{t+1} &= \omega_t + \beta_t(\delta_t - \phi_t^\top\omega_t)\phi_t, \\ \theta_{t+1} &= \theta_t + \alpha_t\delta_t\phi_t - \alpha_t\gamma\phi_t'(\phi_t^\top\omega_t), \end{aligned} \tag{23}$$

where the TD error $\delta_t = r_t + (\gamma\phi_t' - \phi_t)^\top\theta_t$.

Its key matrix is

$$\mathbf{A}_{\text{TDC}} = \mathbf{A}_{\text{off}}^\top\mathbf{C}^{-1}\mathbf{A}_{\text{off}}, \tag{24}$$

$$\mathbf{b}_{\text{TDC}} = \mathbf{A}_{\text{off}}^\top\mathbf{C}^{-1}\mathbf{\Phi}^\top\mathbf{D}_\mu r_\pi, \tag{25}$$

2.2.7. ETD

The update rule of the ETD [2] algorithm is as follows:

$$\begin{aligned}\theta_{t+1} &\doteq \theta_t + \alpha_t F_t \rho_t \Big(r_{t+1} + \gamma \theta_t^\top \boldsymbol{\phi}_{t+1} - \theta_t^\top \boldsymbol{\phi}_t \Big) \boldsymbol{\phi}_t \\ &= \theta_t + \alpha_t \big(F_t \rho_t r_{t+1} \boldsymbol{\phi}_t - F_t \rho_t \boldsymbol{\phi}_t (\boldsymbol{\phi}_t - \gamma \boldsymbol{\phi}_{t+1})^\top \theta_t \big) \\ &= \theta_t + \alpha_t \big(\mathbf{b}_{\mathrm{ETD},t} - \mathbf{A}_{\mathrm{ETD},t} \theta_t \big),\end{aligned} \tag{26}$$

where $F_0 = 1$ and $F_t \doteq \gamma \rho_{t-1} F_{t-1} + 1, \quad \forall t > 0$.

The key matrix of ETD is

$$\begin{aligned}&\mathbf{A}_{\mathrm{ETD}} = \lim_{t\to\infty} \mathbb{E}[\mathbf{A}_{\mathrm{ETD},t}] \\ &= \lim_{t\to\infty} \mathbb{E}_\mu \Big[F_t \rho_t \boldsymbol{\phi}_t (\boldsymbol{\phi}_t - \gamma \boldsymbol{\phi}_{t+1})^\top \Big] \\ &= \sum_s d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu \Big[F_t \rho_t \boldsymbol{\phi}_t (\boldsymbol{\phi}_t - \gamma \boldsymbol{\phi}_{t+1})^\top \mid S_t = s \Big] \\ &= \sum_s \underbrace{d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu[F_t \mid S_t = s]}_{f(s)} \mathbb{E}_\mu \Big[\rho_t \boldsymbol{\phi}_t (\boldsymbol{\phi}_t - \gamma \boldsymbol{\phi}_{t+1})^\top \mid S_t = s \Big] \\ &= \Phi^\top \mathbf{D}_f (\mathbf{I} - \gamma \mathbf{P}_\pi) \Phi,\end{aligned} \tag{27}$$

$$\begin{aligned}\mathbf{b}_{\mathrm{ETD}} &= \lim_{t\to\infty} \mathbb{E}[\mathbf{b}_{\mathrm{ETD},t}] = \lim_{t\to\infty} \mathbb{E}_\mu[F_t \rho_t r_{t+1} \phi_t] \\ &= \Phi^\top \mathbf{D}_f r_\pi,\end{aligned} \tag{28}$$

where $\mathbf{D}_f$ is a diagonal matrix with a diagonal element approximated to $f = (\mathbf{I} - \gamma \mathbf{P}_\pi^\top)^{-1} d_\mu$.

2.2.8. MRetrace

MRetrace [6] is a modified version of Retrace [16] with a convergence guarantee. Its update rule is as follows:

$$\begin{aligned}\theta_{t+1} &\doteq \theta_t + \alpha_t \rho_t \big(r_{t+1} + x_t \gamma \theta_t^\top \phi_{t+1} - \theta_t^\top \phi_t \big) \phi_t \\ &= \theta_t + \alpha_t \big(\rho_t r_{t+1} \phi_t - \rho_t \phi_t (\phi_t - x_t \gamma \phi_{t+1})^\top \theta_t \big) \\ &= \theta_t + \alpha_t \big(\mathbf{b}_{\mathrm{MR},t} - \mathbf{A}_{\mathrm{MR},t} \theta_t \big),\end{aligned} \tag{29}$$

where

$$x_t \doteq \frac{1}{\max_a \rho_t} = \min_a \Big\{ \frac{1}{\rho_t} \Big\} = \min_a \Big\{ \frac{\mu(a|s_t)}{\pi(a|s_t)} \Big\}, \tag{30}$$

$\mathbf{b}_{\mathrm{MR},t} = \rho_t r_{t+1} \phi_t$, $\mathbf{A}_{\mathrm{MR},t} = \rho_t \phi_t (\phi_t - x_t \gamma \phi_{t+1})^\top$.

$$\begin{aligned}\mathbf{b}_{\mathrm{MR}} &= \lim_{t\to\infty} \mathbb{E}[\mathbf{b}_t] = \lim_{t\to\infty} \mathbb{E}_\mu[\rho_t r_{t+1} \phi_t] \\ &= \sum_s d_\mu(s) \mathbb{E}_\mu[\rho_t r_{t+1} \phi_t | S_t = s] \\ &= \sum_s d_\mu(s) \sum_a \mu(a|s) \frac{\pi(a|s)}{\mu(a|s)} r_{t+1} \phi(s) \\ &= \sum_s d_\mu(s) \phi(s) \sum_a \pi(a|s) \sum_{s'} T(s,a,s') R(s,a,s') \\ &= \Phi^\top \mathbf{D}_\mu r_\pi,\end{aligned} \tag{31}$$

The key matrix of MRetrace is

$$
\begin{aligned}
\mathbf{A}_{\mathrm{MR}} &= \lim_{t\to\infty} \mathbb{E}[\mathbf{A}_t] = \lim_{t\to\infty} \mathbb{E}_\mu \Big[\phi_t(\phi_t - x_t\gamma\mathbb{E}_\pi[\phi_{t+1}])^\top\Big] \\
&= \sum_s d_\mu(s)\mathbb{E}_\mu\Big[\phi(s)(\phi(s) - x_t\gamma\mathbb{E}_\pi[\phi(s')])^\top\Big] \\
&= \sum_s d_\mu(s)\phi(s)\big(\phi(s) - \mathbb{E}_\mu[x_t\gamma\mathbb{E}_\pi[\phi(s')]]\big)^\top \\
&= \sum_s d_\mu(s)\phi(s)\Big(\phi(s) - \gamma\sum_a \mu(a|s)\min_b\Big\{\frac{\mu(b|s)}{\pi(b|s)}\Big\}\mathbb{E}_\pi[\phi(s')]\Big)^\top \\
&= \sum_s d_\mu(s)\phi(s)\Big(\phi(s) - \gamma\min_b\Big\{\frac{\mu(b|s)}{\pi(b|s)}\Big\}\sum_{s'}[\mathbf{P}_\pi]_{ss'}\phi(s')\Big)^\top \\
&= \Phi^\top\mathbf{D}_\mu(\mathbf{I} - \gamma\mathbf{D}_x\mathbf{P}_\pi)\Phi,
\end{aligned}
\tag{32}
$$

where $\mathbf{D}_x$ is the $n \times n$ diagonal matrix with d_x on its diagonal, and each component of d_x is $d_x(s) = \min_b\Big\{\frac{\mu(b|s)}{\pi(b|s)}\Big\}$.

3. Finite Sample Analysis

The measurement criteria of finite sample analysis and convergence rate analysis are equivalent. They are both concerned with the relationship between errors and the number of iteration rounds.

3.1. Convergence Rate of General Temporal Difference Learning Algorithm

Let us start with a finite sample analysis of a general temporal difference learning algorithm. For the i.i.d. sequence $\{r_t, \phi_t, \phi'_t\}$, where r_t and ϕ_t are sampled from the Markov process with behavior policy μ, ϕ'_t is sampled from target policy π. Suppose its update rule of parameter θ is defined as follows:

$$\theta_{t+1} = \theta_t + \alpha_t\big(\mathbf{b}_t - \mathbf{A}_t\theta_t\big) = \theta_t + \alpha_t(h(\theta_t) + M_{t+1}), \tag{33}$$

where

$$M_{t+1} = (\mathbf{A} - \mathbf{A}_t)\theta_t + \mathbf{b}_t - \mathbf{b}, \tag{34}$$

$$h(\theta_t) = \mathbf{b} - \mathbf{A}\theta_t = \mathbf{A}\theta^* - \mathbf{A}\theta_t = -\mathbf{A}(\theta_t - \theta^*), \tag{35}$$

where $\mathbf{A} = \lim_{t\to\infty}\mathbb{E}[\mathbf{A}_t]$, $\mathbf{b} = \lim_{t\to\infty}\mathbb{E}[\mathbf{b}_t]$, and the fixed point is $\theta^* = \mathbf{A}^{-1}\mathbf{b}$. $\mathbf{A}$ and $\mathbf{b}$ are based on the i.i.d. sequence $\{r_t, \phi_t, \phi'_t\}$.

Assumption 1. *The key matrix A of the general temporal difference learning algorithm is positive definite.*

Assumption 2. *The sequences $\{r_t,\ \phi_t,\ \phi'_t\}$ have uniformly bounded second moments. Let $\mathcal{F}_t = \sigma(\theta_1, M_1, \ldots, \theta_{t-1}, M_t)$; then, fix some constant $C_s > 0$, such that the following holds:*

$$\mathbb{E}[||M_{t+1}||^2|\mathcal{F}_t] \le C_s(1 + ||\theta_t - \theta^*||^2). \tag{36}$$

This assumption holds for any initial parameter vector θ_1.

Assumption 3. *Step-size sequence α_t satisfies $\alpha_t \in (0,1)$, $\sum_{t=0}^\infty \alpha_t = \infty$, and $\sum_{t=0}^\infty \alpha_t^2 < \infty$.*

Let $\lambda_{\min}(\mathbf{X})$ and $\lambda_{\max}(\mathbf{X})$ denote the minimum and maximum eigenvalues of the matrix $\mathbf{X}$.

Theorem 1. *(Convergence Rate in Expectation for General Temporal Difference Learning Algorithm). Assume that Assumptions 1–3 hold. For $t \geq 0$, we have*

$$\mathbb{E}\|\theta_{t+1} - \theta^*\|^2 \leq e^{\lambda_0^t}\mathbb{E}\|\theta_0 - \theta^*\|^2 + C_s \sum_{i=0}^{t} \left[e^{\lambda_{i+1}^t}\right] \alpha_i^2, \tag{37}$$

where

$$\lambda_i^t = \begin{cases} -\lambda_{\min}(\boldsymbol{A} + \boldsymbol{A}^\top) \sum_{k=i}^t \alpha_k + \lambda_{\max}(\boldsymbol{A}^\top \boldsymbol{A} + C_s \boldsymbol{I}) \sum_{k=i}^t \alpha_k^2, & i \leq t, \\ 0, & i > t. \end{cases} \tag{38}$$

Proof. Note that the proof process is similar to the proof of Theorem 3.1 of [7].

Based on Definitions (33) and (35), we have

$$\begin{aligned} \theta_{t+1} - \theta^* &= \theta_t + \alpha_t(h(\theta_t) + M_{t+1}) - \theta^* \\ &= \theta_t - \theta^* + \alpha_t(-\mathbf{A}(\theta_t - \theta^*) + M_{t+1}) \\ &= (\mathbf{I} - \alpha_t\mathbf{A})(\theta_t - \theta^*) + \alpha_t M_{t+1}. \end{aligned} \tag{39}$$

$$\begin{aligned} \|\theta_t - \theta^*\|^2 =& (\theta_{t+1} - \theta^*)^\top (\theta_{t+1} - \theta^*) \\ =& [(\mathbf{I} - \alpha_t\mathbf{A})(\theta_t - \theta^*) + \alpha_t M_{t+1}]^\top [(\mathbf{I} - \alpha_t\mathbf{A})(\theta_t - \theta^*) + \alpha_t M_{t+1}] \\ =& (\theta_t - \theta^*)^\top (\mathbf{I} - \alpha_t\mathbf{A})^\top (\mathbf{I} - \alpha_t\mathbf{A})(\theta_t - \theta^*) \\ &+ \alpha_t(\theta_t - \theta^*)^\top (\mathbf{I} - \alpha_t\mathbf{A})^\top M_{t+1} + \alpha_t M_{t+1}^\top (\mathbf{I} - \alpha_t\mathbf{A})(\theta_t - \theta^*) \\ &+ \alpha_t^2 \|M_{t+1}\|^2. \end{aligned} \tag{40}$$

Taking conditional expectations on both sides, and using $\mathbb{E}[M_{t+1}|\mathcal{F}_t] = 0$, we get

$$\mathbb{E}[\|\theta_{t+1} - \theta^*\|^2|\mathcal{F}_t] = \alpha_t^2\mathbb{E}[\|M_{t+1}\|^2|\mathcal{F}_t] + (\theta_t - \theta^*)^\top (\mathbf{I} - \alpha_t\mathbf{A})^\top (\mathbf{I} - \alpha_t\mathbf{A})(\theta_t - \theta^*). \tag{41}$$

Therefore, using Assumption 2,

$$\mathbb{E}[\|\theta_{t+1} - \theta^*\|^2|\mathcal{F}_t] \leq (\theta_t - \theta^*)^\top \Lambda_t(\theta_t - \theta^*) + C_s\alpha_t^2, \tag{42}$$

where

$$\Lambda_t = (\mathbf{I} - \alpha_t\mathbf{A})^\top (\mathbf{I} - \alpha_t\mathbf{A}) + C_s\alpha_t^2\mathbf{I}. \tag{43}$$

Since Λ_t is a symmetric matrix and the sum of two positive definite matrices, all its eigenvalues are real and positive. Let $\lambda_t := \lambda_{\max}(\Lambda_t)$; thus, we have $\lambda_t > 0$ and

$$\mathbb{E}[\|\theta_{t+1} - \theta^*\|^2|\mathcal{F}_t] \leq \lambda_t\|\theta_t - \theta^*\|^2 + C_s\alpha_t^2. \tag{44}$$

Taking the expectations on both sides, we have

$$\mathbb{E}\|\theta_{t+1} - \theta^*\|^2 \leq \lambda_t\mathbb{E}\|\theta_t - \theta^*\|^2 + C_s\alpha_t^2. \tag{45}$$

Sequentially using the above inequality, we have

$$\mathbb{E}\|\theta_{t+1} - \theta^*\|^2 \leq \left[\prod_{k=0}^{t} \lambda_k\right] \mathbb{E}\|\theta_0 - \theta^*\|^2 + C_s \sum_{i=0}^{t} \left[\prod_{k=i+1}^{t} \lambda_k\right] \alpha_t^2. \tag{46}$$

where we let $\prod_{k=t+1}^t \lambda_k = 1$.

Based on Assumption 1, $\mathbf{A}$ is positive definite, and the matrices $(\mathbf{A}+\mathbf{A}^\top)$ and $(\mathbf{A}^\top\mathbf{A}+C_s\mathbf{I})$ in (38) are positive definite. Thus, their minimum and maximum eigenvalues are strictly positive. Hence, using Weyl's inequality, we have

$$\begin{aligned} \lambda_t &= \lambda_{\max}((\mathbf{I}-\alpha_t\mathbf{A})^\top(\mathbf{I}-\alpha_t\mathbf{A})+C_s\alpha_t^2\mathbf{I}) \\ &\le \lambda_{\max}(\mathbf{I}-\alpha_t(\mathbf{A}+\mathbf{A}^\top))+\alpha_t^2\lambda_{\max}(\mathbf{A}^\top\mathbf{A}+C_s\mathbf{I}) \\ &\le 1-\alpha_t\lambda_{\min}(\mathbf{A}+\mathbf{A}^\top)+\alpha_t^2\lambda_{\max}(\mathbf{A}^\top\mathbf{A}+C_s\mathbf{I}) \\ &\le e^{\left[-\alpha_t\lambda_{\min}(\mathbf{A}+\mathbf{A}^\top)+\alpha_t^2\lambda_{\max}(\mathbf{A}^\top\mathbf{A}+C_s\mathbf{I})\right]} \end{aligned} \tag{47}$$

In the case of $\lambda_t > 0, \forall t > 0$, when $0 < i < t$, for the concatenated multiplication of (47) from i to t, we have

$$\begin{aligned} \prod_{k=i}^{t}\lambda_k &\le \prod_{k=i}^{t} e^{\left[-\alpha_k\lambda_{\min}(\mathbf{A}+\mathbf{A}^\top)+\alpha_k^2\lambda_{\max}(\mathbf{A}^\top\mathbf{A}+C_s\mathbf{I})\right]} \\ &= e^{\sum_{k=i}^{t}\left[-\alpha_k\lambda_{\min}(\mathbf{A}+\mathbf{A}^\top)+\alpha_k^2\lambda_{\max}(\mathbf{A}^\top\mathbf{A}+C_s\mathbf{I})\right]} \\ &= e^{\lambda_i^t} \end{aligned} \tag{48}$$

The claim now follows. □

3.2. Convergence Rate of the MRetrace Algorithm

Assumption 4. *The Markov Chain (s_t) is aperiodic and irreducible; thus, $\lim_{t\to\infty}\mathbb{P}(s_t = s'|s_0 = s) = d_\mu(s')$ exists and is unique.*

This assumption implies that the state distribution vector d_μ of the behavior policy μ is the fixed point of

$$d_\mu = \mathbf{P}_\mu^\top d_\mu, \tag{49}$$

where the element of matrix $\mathbf{P}_\mu$ is as follows:

$$[\mathbf{P}_\mu]_{ss'} = \sum \mu(a|s)\mathcal{T}(s,a,s'). \tag{50}$$

Assumption 5. *$\{\phi_t, r_t, \mathbb{E}_\pi[\phi_{t+1}]\}$ is such that $\mathbb{E}_\mu[||\phi_t||^2|s_{t_1}]$, $\mathbb{E}_\mu[r_t^2|s_{t_1}]$, $\mathbb{E}_\pi[||\phi_{t+1}||^2|s_{t_1}]$ are uniformly bounded.*

Assumption 6. *The feature matrix Φ is column full rank.*

Corollary 1. *(Convergence Rate in Expectation for MRetrace algorithm). Assume Assumptions 4–6 hold. Fix some constant $C_s > 0$, for $t \ge 0$, we have*

$$\mathbb{E}||\theta_{t+1}-\theta^*||^2 \le e^{\lambda_0^t}\mathbb{E}||\theta_0-\theta^*||^2 + C_s\sum_{i=0}^{t}\left[e^{\lambda_{i+1}^t}\right]\alpha_i^2. \tag{51}$$

where

$$\lambda_i^t = \begin{cases} -\lambda_{\min}(\boldsymbol{A}_{MR}+\boldsymbol{A}_{MR}^\top)\sum_{k=i}^{t}\alpha_k + \lambda_{\max}(\boldsymbol{A}_{MR}^\top\boldsymbol{A}_{MR}+C_s\boldsymbol{I})\sum_{k=i}^{t}\alpha_k^2, & i \le t, \\ 0, & i > t. \end{cases} \tag{52}$$

Proof. All we need is to show that the MRetrace algorithm satisfies the assumptions of Theorem 1.

According to the proof of Theorem 1 of [6], given Assumptions 4 and 6, the matrix $\mathbf{A}_{\mathrm{MR}}$ is positive definite. Under Assumption 5, there exists some constant $C_s > 0$, $\mathbb{E}[||M_{t+1}||^2|\mathcal{F}_t] \le C_s(1+||\theta_t-\theta^*||^2)$. Then, based on Assumption 3, the claim follows by directly applying Theorem 1. □

4. How to Compare?

Theorem 1 and Corollary 1 are essentially simplified versions of Theorem 3.1 of [7], but their advantage lies in facilitating the analysis of the main factor affecting convergence rates.

To ensure a fair comparison among different learning algorithms, we need the same setting for each algorithm.

Assumption 7. *Assume each algorithm shares the same feature matrix, the same behavior policy, the same target policy, the same initial parameters θ_0, the same constant C_s, and the same learning rate sequences α_t.*

Corollary 2. *(The main factor affecting convergence rates). Assume Assumptions 1–3, 6 and 7. From the perspective of the expected convergence rate, the main factor that affects the convergence rate is the minimum eigenvalue $\frac{1}{2}\lambda_{\min}(\boldsymbol{A}+\boldsymbol{A}^\top)$ of the key matrix $\boldsymbol{A}$. Furthermore, the larger the minimum eigenvalue of the key matrix, the faster the algorithm has a convergence rate. (This corollary is not actually what we discovered first. An anonymous expert reviewer of UAI2023 pointed out the role of the smallest eigenvalue. However, we did not find any relevant evidence or conclusions in the existing literature, so we formally stated this conclusion here).*

Proof. Based on Assumptions 1 and 2, we obtain Theorem 1 on the expected convergence rate. Checking the error bound (37) in Theorem 1, removing the identical settings for different algorithms, one can easily find that each term contains only a key term $e^{\lambda_i^t}$, where $i \in [0, t]$. When $i < t$,

$$\lambda_i^t = -\lambda_{\min}(\mathbf{A}+\mathbf{A}^\top)\sum_{k=i}^{t}\alpha_k + \lambda_{\max}(\mathbf{A}^\top\mathbf{A}+C_s\mathbf{I})\sum_{k=i}^{t}\alpha_k^2. \tag{53}$$

Based on Assumption 3, we have $\sum_{k=i}^{t}\alpha_t > \sum_{k=i}^{t}\alpha_t^2$. Furthermore, fix variable i; thus, we have

$$\lim_{t\to\infty}\sum_{k=i}^{t}\alpha_t = \infty, \tag{54}$$

and

$$\lim_{t\to\infty}\sum_{k=i}^{t}\alpha_t^2 < \infty. \tag{55}$$

That is

$$\lim_{t\to\infty}\sum_{k=i}^{t}\alpha_k \gg \lim_{t\to\infty}\sum_{k=i}^{t}\alpha_k^2. \tag{56}$$

Therefore, the main factor in (53) is $-\lambda_{\min}(\mathbf{A}+\mathbf{A}^\top)\sum_{k=i}^{t}\alpha_k$. Finally, based on the same learning rate sequence, $\frac{1}{2}\lambda_{\min}(\mathbf{A}+\mathbf{A}^\top)$ is the main factor that affects the convergence rate.

In (37), for a given t, a smaller bound indicates a faster convergence rate. In (37), all elements including all $e^{\lambda_i^t}$, $\mathbb{E}\|\theta_0-\theta^*\|^2$, C_s, and all α_i, are greater than zero, meaning that a smaller $e^{\lambda_i^t}$ leads to a faster convergence rate. Therefore, the smaller the value of λ_i^t in (53), the faster the convergence rate. Hence, the larger the value of $\frac{1}{2}\lambda_{\min}(\mathbf{A}+\mathbf{A}^\top)$, the faster the convergence rate. □

5. Numerical Analysis

To compare the expected convergence rates of various algorithms, what we are going to do next is to compute and compare the minimum eigenvalues of each algorithm in different environments and different policies based on Corollary 2. The environments include two-state counterexample [2], Baird's counterexample [3,5], Random Walk with tabular feature [5], Random Walk with inverted feature [5], Random Walk with dependent feature [5], and Boyan Chain [5,17]. Furthermore, in Random Walk, the target policy takes the right action in 60% of the time and the behavior policy selects the right and left action with equal probability [18]. In the Boyan Chain, the target policy and the behavior policy are the same.

5.1. Example Settings

Two-state counterexample: The $\theta \to 2\theta$ problem has only two states. From each state, there are two actions, *left* and *right*, which take the agent to the left or right state. All rewards are zeros. The features $\Phi = (1, 2)^\top$ are assigned to the left and the right state. The schematic of the environment is shown in Figure 1.

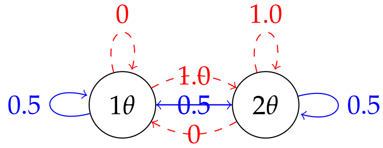

Figure 1. Two-state counterexample, where the blue solid arrows represent the behavior policy μ, and the red dashed arrows represent the target policy π.

The behavior policy takes the equal probability to *left* or *right* in both states, i.e., $\mathrm{P}_\mu = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$.

The target policy only selects action right in both states, i.e., $\mathrm{P}_\pi = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$.

The state distribution $d_\mu = (0.5, 0.5)^\top$, $d_x = d_c = (0.5, 0.5)^\top$, $\mathbf{C} = 2.5$, $f = (0.5, 9.5)^\top$, $r_\pi = 0$, $r_c = 0$.

Baird's counterexample: Baird's counterexample has seven states, and the schematic of the environment is shown in Figure 2.

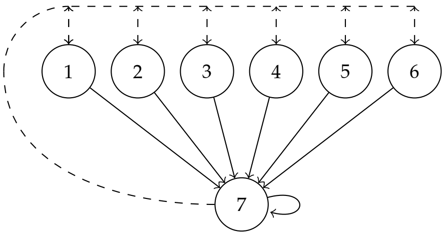

Figure 2. Baird's counterexample, where the probability of the *solid* action and the *dashed* action in behavior policy μ and target policy π are $\mu(\mathit{dashed} \mid \cdot) = \frac{6}{7}$, $\mu(\mathit{solid} \mid \cdot) = \frac{1}{7}$ and $\pi(\mathit{solid} \mid \cdot) = 1$.

The feature matrix is of dimensions (7×8) where each state is represented by an 8-dimensional feature, i.e.,

$$\Phi = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 \end{bmatrix}.$$

The behavior policy takes equal probabilities to each state, i.e.,

$$\mathrm{P}_{\mu} = \begin{bmatrix} \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \end{bmatrix}.$$

The target policy only selects the action to state 7 in both states, i.e.,

$$\mathrm{P}_{\pi} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The state distribution

$$d_{\mu} = (\frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7})^{\top},$$

$$d_{x} = d_{c} = (\frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7})^{\top},$$

$$\mathbf{C} = \begin{bmatrix} 0.571 & 0 & 0 & 0 & 0 & 0 & 0.286 \\ 0 & 0.571 & 0 & 0 & 0 & 0 & 0.286 \\ 0 & 0 & 0.571 & 0 & 0 & 0 & 0.286 \\ 0 & 0 & 0 & 0.571 & 0 & 0 & 0.286 \\ 0 & 0 & 0 & 0 & 0.571 & 0 & 0.286 \\ 0 & 0 & 0 & 0 & 0 & 0.571 & 0.286 \\ 0.286 & 0.286 & 0.286 & 0.286 & 0.286 & 0.286 & 1.429 \end{bmatrix}.$$

$f = (\frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{64}{7})^{\top}$, $r_{\pi} = (0,0,0,0,0,0,0)^{\top}$, $r_{c} = (0,0,0,0,0,0,0)^{\top}$.

Random Walk: Random Walk is centered around a typical Markov Chain. This chain comprises five consecutive states, with two terminal states positioned at each extremity serving as absorptive endpoints.

The schematic of the environment is shown in Figure 3.

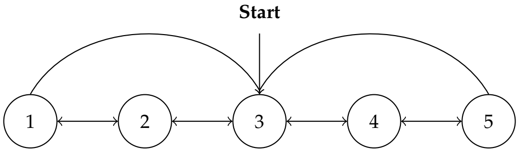

Figure 3. Random Walk. All walks begin in state 3. Under the behavior policy, take either a *left* or *right* action with a probability of 0.5 in each state. Under the target policy, take either a *left* or *right* action with a probability of 0.4 or 0.6 in each state.

Specifically, its state–state transfer probability of behavior policy is

$$\mathrm{P}_{\mu} = \begin{bmatrix} 0 & 0.6 & 0.4 & 0 & 0 \\ 0.4 & 0 & 0.6 & 0 & 0 \\ 0 & 0.4 & 0 & 0.6 & 0 \\ 0 & 0 & 0.4 & 0 & 0.6 \\ 0 & 0 & 0.6 & 0.4 & 0 \end{bmatrix}.$$

The state–state transfer probability of the target policy is

$$\mathrm{P}_{\pi} = \begin{bmatrix} 0 & 0.5 & 0.5 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0 & 0.5 \\ 0 & 0 & 0.5 & 0.5 & 0 \end{bmatrix}.$$

The state distribution $d_{\mu} = (\frac{1}{9}, \frac{2}{9}, \frac{3}{9}, \frac{2}{9}, \frac{1}{9})^{\top}$, $d_c = (0.9, 0.9, 0.9, 0.9, 0.9)^{\top}$, $d_c = (\frac{5}{6}, \frac{5}{6}, \frac{5}{6}, \frac{5}{6}, \frac{5}{6})^{\top}$, $f = (0.773, 1.84, 3.333, 2.56, 1.493)^{\top}$, $r_{\pi} = (-0.4, 0, 0, 0, 0.6)^{\top}$, $r_c = (-0.4, 0, 0, 0, 0.5)^{\top}$.

The Random Walk environment has three feature representations, which are called tabular features, inverted features, and dependent features. The feature matrix of Random Walk with tabular features is

$$\Phi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$\mathbf{C} = \begin{bmatrix} 0.111 & 0 & 0 & 0 & 0 \\ 0 & 0.222 & 0 & 0 & 0 \\ 0 & 0 & 0.333 & 0 & 0 \\ 0 & 0 & 0 & 0.222 & 0 \\ 0 & 0 & 0 & 0 & 0.111 \end{bmatrix}.$$

The feature matrix of Random Walk with inverted features is

$$\Phi = \begin{bmatrix} 0 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0 \end{bmatrix}.$$

$$\mathbf{C} = \begin{bmatrix} 0.222 & 0.167 & 0.139 & 0.167 & 0.194 \\ 0.167 & 0.194 & 0.111 & 0.139 & 0.167 \\ 0.139 & 0.111 & 0.167 & 0.111 & 0.139 \\ 0.167 & 0.139 & 0.111 & 0.194 & 0.167 \\ 0.194 & 0.167 & 0.139 & 0.167 & 0.222 \end{bmatrix}.$$

The feature matrix of Random Walk with dependent features is

$$\Phi = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & 1 \end{bmatrix}.$$

$$\mathbf{C} = \begin{bmatrix} 0.333 & 0.222 & 0.111 \\ 0.222 & 0.333 & 0.222 \\ 0.111 & 0.222 & 0.333 \end{bmatrix}.$$

Boyan Chain: The Boyan Chain consists of 13 states, each of which is represented by 4 state features. The feature matrix of the Boyan Chain with dependent features is

$$\Phi = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.75 & 0.25 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \\ 0.25 & 0.75 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0.75 & 0.25 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0.25 & 0.75 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0.75 & 0.25 \\ 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0.25 & 0.75 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The state distribution $d_\mu = (0.108, 0.054, 0.081, 0.068, 0.075, 0.071, 0.073, 0.072, 0.072, 0.072, 0.072, 0.072, 0.108)^\top$, $d_x = d_c = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)^\top$,

$$\mathbf{C} = \begin{bmatrix} 0.163 & 0.043 & 0 & 0 \\ 0.043 & 0.199 & 0.045 & 0 \\ 0 & 0.045 & 0.199 & 0.045 \\ 0 & 0 & 0.045 & 0.172 \end{bmatrix}.$$

$f = (1.084, 0.542, 0.813, 0.678, 0.745, 0.712, 0.729, 0.72, 0.724, 0.722, 0.723, 0.723, 1.084)^\top$, $r_\pi = r_c = (-3, -3, -3, -3, -3, -3, -3, -3, -3, -3, -3, -2, 0)^\top$.

Then, the state–state transfer probability of behavior policy is the same as the state–state transfer probability of target policy, i.e., $\mathrm{P}_\mu = \mathrm{P}_\pi$, where the element of matrix P_μ is as follows, for $i, j \in [0, 12]$:

$$P_\mu[i,j] = \begin{cases} 0.5 & \text{if } i \leq 10 \text{ and } j = i+1 \\ 0.5 & \text{if } i \leq 10 \text{ and } j = i+2 \\ 1 & \text{if } i = 11 \text{ and } j = 12 \\ 1 & \text{if } i = 12 \text{ and } j = 0 \\ 0 & \text{otherwise.} \end{cases}$$

The schematic of the environment is shown in Figure 4.

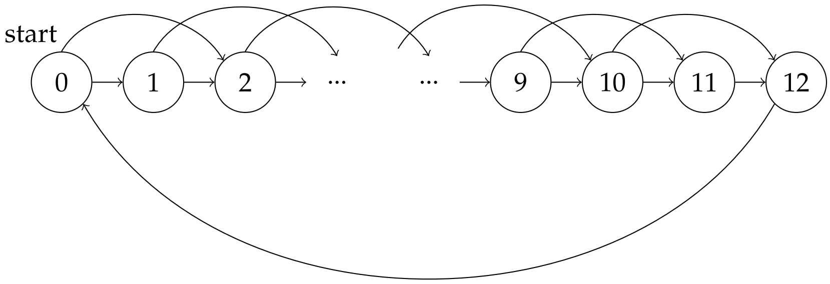

Figure 4. Boyan Chain. In state 0–10, each *solid* action is taken with probability 0.5. In states 11 and 12, the probability of a *solid* action is 1.

5.2. Results and Analysis

Based on Table 1, first, we need to set Φ, P_μ, P_π, D_c, r_c, r_π, C, D_f, and D_x for each setting. Then, based on the property $d_\mu = \mathbf{P}_\mu^\top d_\mu$, we compute the eigenvector of the matrix $\mathbf{P}_\mu^\top$ with eigenvalue 1.0, and we unitize it to obtain d_μ and D_μ. (It is important to note that when there are absorbing states in a Markov Chain, the probability distribution is 1.0 only at the absorbing states and 0 at all other states. Therefore, we adopt a restart approach, jumping directly back to the initial states, thus forming a probability transition matrix $\mathbf{P}_\mu$ without absorbing states.) After all, we compute the key matrix for each algorithm and its minimum eigenvalue. Note that the step-size ratio η of the auxiliary parameter to the learning parameter is usually set to $\eta \geq 1.0$.

The minimum eigenvalues in several examples for each algorithm are summarized in Table 2. We can find the following: (1) In Baird's counterexample and a two-state counterexample, the minimum eigenvalues of key matrices in off-policy TD and Retrace are both less than 0, indicating that they will diverge in these two counterexamples, which is consistent with the existing research [2]. Additionally, compared with off-policy TD, Retrace does have some mitigation towards divergence but cannot avoid it.

Table 2. Minimum eigenvalues $\frac{1}{2}\lambda_{\min}(\mathbf{A} + \mathbf{A}^\top)$ of various algorithms on several examples.

Algorithm	**Two-State**	**Baird's**	**Random Walk**			**Boyan Chain**
			Tabular	**Inverted**	**Dependent**	
Off-policy TD	−0.2	−0.791	0.018	0.017	0.07	0.024
Retrace	−0.1	−0.113	0.017	0.015	0.063	0.024
BR	0.34	$\mathbf{9.673 \times 10^{-17}}$	0.002	0.007	0.033	0.002
GTD	0	0	0	0	0	0
GTD2	0	-1.077×10^{-17}	0	0	0	0
TDC	0.016	−0.002	0.002	0.007	0.011	0.002
ETD	**3.4**	-2.82×10^{-16}	**0.195**	**0.165**	**0.76**	**0.245**
MRetrace	1.15	-2.141×10^{-17}	0.046	0.02	0.094	0.024

(2) In Baird's counterexample, the minimum eigenvalues of GTD2, TDC, ETD, and MRetrace are all less than zero, indicating that their key matrices are not positive definite. This seems inconsistent with our understanding. The main reason is that in Baird's counterexample, the feature matrix is 7 × 8, and is not column full rank, which is inconsistent with the assumption in their theorems. Note that the absolute value of the minimum eigenvalue is very small.

(3) Except for Baird's counterexample, the minimum eigenvalues of GTD and GTD2 are all zeros. The reason is that matrices $\frac{1}{2}\lambda_{\min}(\mathbf{A} + \mathbf{A}^\top)$ of GTD and GTD2 are positive semi-definite. In the context of this paper, we are unable to distinguish which one is faster between GTD and GTD2 in numerical analysis.

(4) The minimum eigenvalue of TDC is higher than GTD and GTD2, which is consistent with the literature.

(5) Except for Baird's counterexample, the minimum eigenvalue of ETD is the largest.

(6) Except for Baird's counterexample, MRetrace has the second-largest minimum eigenvalues.

(7) All the minimum eigenvalues of BR are greater than 0, making it the only one to remain positive definite in all examples. However, except for Baird's counterexample, its minimum eigenvalue is not large.

(8) The Boyan Chain is an on-policy setting; off-policy TD is implemented in on-policy; and off-policy TD, Retrace, and MRetrace have the same minimum eigenvalue of 0.024. BR and TDC have the same minimum eigenvalue of 0.002, indicating that BR and TDC are not suitable for on-policy learning.

(9) It is surprising that in an on-policy setting, the minimum eigenvalue of ETD is larger than that of on-policy TD. This implies that in terms of the expected convergence rate, ETD is the most recommended option for on-policy learning.

To the best of our knowledge, this is the first time that various temporal difference learning algorithms have been compared for their convergence rates in a very intuitive numerical analysis manner.

In summary, in the expected sense, ETD has the fastest convergence rate, followed by MRetrace.

6. Experimental Studies

In Section 3, we proposed Corollary 2, and in Section 4, we compared the sizes of the minimum eigenvalues of different algorithms for various environment settings. This means that we carried out theoretical analysis combined with numerical analysis, but whether the analytical results reflect the actual situation needs to be experimentally verified. Therefore, we adopted the same environment settings as in Section 4.

Each algorithm runs independently 20 times, with 1000 steps per run, and calculates the mean and standard deviation.

To compare convergence rates, we need to observe the trend of $|\theta_t - \theta^*|$ over time step t. Note that according to Table 1, different algorithms have different optimal solutions $\theta^* = \mathbf{A}^{-1}\mathbf{b}$, so we need to calculate their optimal solutions separately for each algorithm and then measure the errors.

The learning rate is set to satisfy Assumption 3, $\alpha_t = \alpha_0 \times \frac{1}{t+1}$, where α_0 is an initial learning rate. We set $\eta = 4$ for GTD and GTD2 in all environments. In two-state counterexamples, $\alpha_0 = 0.1$, and $\gamma = 0.9$. In Baird's counterexamples, $\alpha_0 = 0.05$, and $\gamma = 0.99$. In Random Walk, $\alpha_0 = 0.25$, and $\gamma = 0.9$. In a Boyan Chain, $\alpha_0 = 0.25$, and $\gamma = 0.9$.

The learning curves of different algorithms in different environments are shown in Figure 5, where each curve displays the mean and standard deviation of the errors. We can find the following: (1) In Random Walk with dependent features, the convergence rate analysis is consistent with the learning curves of each algorithm. They have the same order: ETD $>$ MRetrace > off-policy TD > Retrace > TDC $\geq$ BR > GTD2 $\geq$ GTD.

(2) Off-policy TD and Retrace diverge in both counterexamples, but Retrace diverges more slowly compared with off-policy TD. This is consistent with the numerical analysis in Section 4.

(3) In Baird's counterexample, GTD descends faster than GTD2 and TDC. This may be related to the numerical analysis showing that the minimum eigenvalues of GTD2 and TDC are less than 0. Therefore, this is consistent with the analysis.

(4) In Baird's counterexample, ETD diverges. On the one hand, this is related to high variances of ETD reported in the literature [6,19]; on the other hand, it is also related to the numerical analysis, showing that the minimum eigenvalue is less than 0.

(5) In a two-state counterexample, compared to BR, ETD, and MRetrace, the algorithms GTD, GTD2, and TDC converge very slowly. Additionally, BR converges slower than MRetrace and ETD. These observations are consistent with our numerical analysis. However, ETD is slower than MRetrace, which is attributed to the high variance of ETD.

(6) In Random Walk and the Boyan Chain, TDC converges much faster than GTD and GTD2, which is consistent with our numerical analysis. This observation also aligns with existing research.

(7) In Random Walk and the Boyan Chain, the convergence rate of ETD is remarkably fast, which aligns with our numerical analysis. ETD is reported to be suitable for off-policy learning, but its exceptional performance in on-policy learning is somewhat incredible.

(8) In the Boyan Chain, off-policy TD is implemented as on-policy TD. The learning curves of TD, Retrace, and MRetrace overlap. This is consistent with our numerical analysis.

In conclusion, the convergence rates of the algorithms in the experiments align with the numerical analysis. Moreover, any inconsistencies have interpretable underlying reasons.

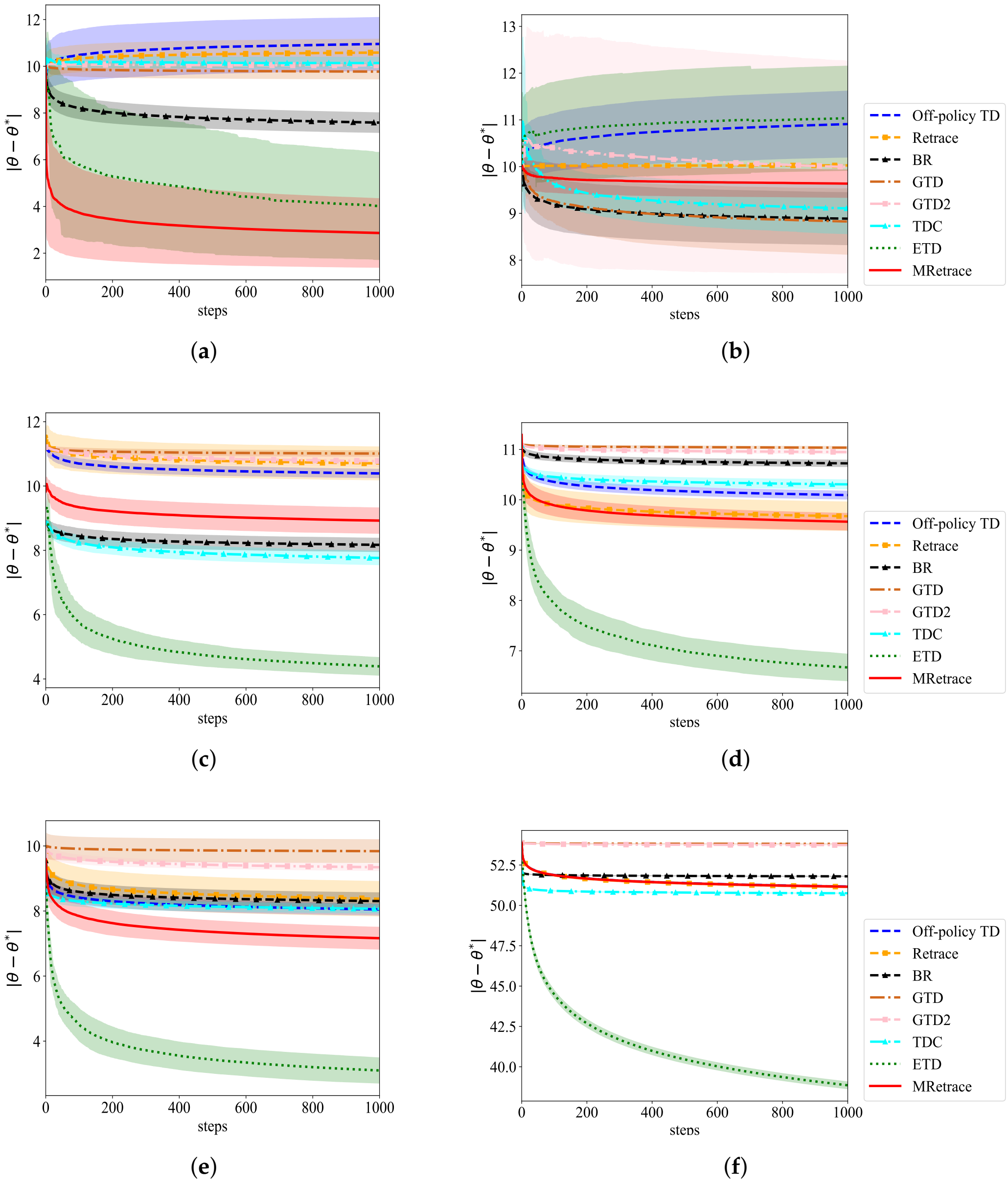

Figure 5. Comparisons of learning curves in different environments. (**a**) Two-state counterexamples; (**b**) Baird's counterexamples; (**c**) Random Walk with tabular features; (**d**) Random Walk with inverted features; (**e**) Random Walk with dependent features; (**f**) Boyan Chain.

7. Conclusions and Future Work

Based on the proposed convergence rate for constructing general off-policy temporal difference learning algorithms, this paper proved that the primary determinant influencing convergence rate is the minimum eigenvalue of the key matrix. Focusing on this factor will be more conducive to the development of faster converging off-policy learning algorithms.

The limitations of this paper include the following aspects:

(1) This paper assumes that the learning rates of all algorithms are the same; however, in reality, different algorithms have different ranges of applicable learning rates.
(2) This paper does not consider the scenario of a fixed learning rate.
(3) This paper focuses on learning prediction and does not address learning control.

Future works need to address the above limitations and explore how to design faster algorithms based on the conclusions of this paper.

In the end, we discovered a contradiction. Sutton et al. [4] proved that $\mathbf{A}_{\text{GTD}} = \begin{pmatrix} \sqrt{\eta}\mathbf{I} & \mathbf{A}_{\text{off}} \\ -\mathbf{A}_{\text{off}}^{\top} & 0 \end{pmatrix}$ is positive definite. Note that in the paper, $\mathbf{G} = \begin{pmatrix} -\sqrt{\eta}\mathbf{I} & -\mathbf{A}_{\text{off}} \\ \mathbf{A}_{\text{off}}^{\top} & 0 \end{pmatrix}$ is proved to be negative definite [4]. According to Theorem A.3 of [20], the positive definiteness of square matrix $\mathbf{A}$ is equivalent to the positive definiteness of $(\mathbf{A} + \mathbf{A}^{\top})$; thus, $(\mathbf{A}_{\text{GTD}} + \mathbf{A}_{\text{GTD}}^{\top})$ is positive definite. However, our calculations in this paper showed that $(\mathbf{A}_{\text{GTD}} + \mathbf{A}_{\text{GTD}}^{\top}) = \begin{pmatrix} 2\sqrt{\eta}\mathbf{I} & 0 \\ 0 & 0 \end{pmatrix}$ is positive semi-definite, not positive definite. Therefore, the conclusion that $\mathbf{A}_{\text{GTD}}$ is positive definite is questionable.

Author Contributions: Conceptualization, X.C.; methodology, X.C.; software, W.Q.; formal analysis, W.Q.; investigation, Y.G.; discussion, S.Y.; writing—review and editing, X.C.; supervision, W.W. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the National Natural Science Foundation, China (Nos. 62276142, 62206133, and 62202240).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of this study; in the collection, analyses, or interpretation of data; in the writing of this manuscript; or in the decision to publish the results.

References

1. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; The MIT Press: Cambridge, MA, USA, 2018.
2. Sutton, R.S.; Mahmood, A.R.; White, M. An emphatic approach to the problem of off-policy temporal-difference learning. *J. Mach. Learn. Res.* **2016**, *17*, 2603–2631.
3. Baird, L. Residual algorithms: Reinforcement learning with function approximation. In Proceedings of the 12th International Conference on Machine Learning, Tahoe City, CA, USA, 9–12 July 1995; pp. 30–37.
4. Sutton, R.S.; Maei, H.R.; Szepesvári, C. A Convergent $O(n)$ Temporal-difference Algorithm for Off-policy Learning with Linear Function Approximation. *Adv. Neural Inf. Process. Syst.* **2008**, *21*, 1609–1616.
5. Sutton, R.; Maei, H.; Precup, D.; Bhatnagar, S.; Silver, D.; Szepesvári, C.; Wiewiora, E. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In Proceedings of the 26th International Conference on Machine Learning, Montreal, QC, Canada, 14–18 June 2009; pp. 993–1000.
6. Chen, X.; Ma, X.; Li, Y.; Yang, G.; Yang, S.; Gao, Y. Modified retrace for off-policy temporal difference learning. In Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence, Pittsburgh, PA, USA, 31 July–4 August 2023; pp. 303–312.
7. Dalal, G.; Szörényi, B.; Thoppe, G.; Mannor, S. Finite sample analyses for TD (0) with function approximation. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LO, USA, 2–7 February 2018; Volume 32.
8. Dalal, G.; Thoppe, G.; Szörényi, B.; Mannor, S. Finite sample analysis of two-timescale stochastic approximation with applications to reinforcement learning. In Proceedings of the Conference On Learning Theory, PMLR, Stockholm, Sweden, on 5–9 July 2018; pp. 1199–1233.
9. Gupta, H.; Srikant, R.; Ying, L. Finite-time performance bounds and adaptive learning rate selection for two time-scale reinforcement learning. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 4704–4713.
10. Xu, T.; Zou, S.; Liang, Y. Two time-scale off-policy TD learning: Non-asymptotic analysis over Markovian samples. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 10634–10644.
11. Dalal, G.; Szorenyi, B.; Thoppe, G. A tale of two-timescale reinforcement learning with the tightest finite-time bound. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–8 February 2020; Volume 34, pp. 3701–3708.
12. Durmus, A.; Moulines, E.; Naumov, A.; Samsonov, S.; Scaman, K.; Wai, H.T. Tight high probability bounds for linear stochastic approximation with fixed stepsize. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 30063–30074.
13. Xu, T.; Liang, Y. Sample complexity bounds for two timescale value-based reinforcement learning algorithms. In Proceedings of the International Conference on Artificial Intelligence and Statistics, PMLR, Virtual, 13–15 April 2021; pp. 811–819.
14. Zhang, S.; Des Combes, R.T.; Laroche, R. On the convergence of SARSA with linear function approximation. In Proceedings of the International Conference on Machine Learning, PMLR, Honolulu, HI, USA, 23–29 July 2023; pp. 41613–41646.

15. Wang, S.; Si, N.; Blanchet, J.; Zhou, Z. A finite sample complexity bound for distributionally robust q-learning. In Proceedings of the International Conference on Artificial Intelligence and Statistics, PMLR, Honolulu, HI, USA, 23–29 July 2023; pp. 3370–3398.
16. Munos, R.; Stepleton, T.; Harutyunyan, A.; Bellemare, M. Safe and efficient off-policy reinforcement learning. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 1054–1062.
17. Boyan, J.A. Technical update: Least-squares temporal difference learning. *Mach. Learn.* **2002**, *49*, 233–246. [CrossRef]
18. Ghiassian, S.; Patterson, A.; Garg, S.; Gupta, D.; White, A.; White, M. Gradient temporal-difference learning with regularized corrections. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 13–18 July 2020; pp. 3524–3534.
19. Zhang, S.; Whiteson, S. Truncated emphatic temporal difference methods for prediction and control. *J. Mach. Learn. Res.* **2022**, *23*, 6859–6917.
20. Sutton, R.S. Learning to predict by the methods of temporal differences. *Mach. Learn.* **1988**, *3*, 9–44. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.