Modified Retrace for Off-Policy Temporal Difference Learning

Abstract

Off-policy learning is a key to extend reinforcement learning as it allows to learn a target policy from a different behavior policy that generates the data. However, it is well known as "the deadly triad" when combined with bootstrapping and function approximation. Retrace is an efficient and convergent off-policy algorithm with tabular value functions which employs truncated importance sampling ratios. Unfortunately, Retrace is known to be unstable with linear function approximation. In this paper, we propose modified Retrace to correct the off-policy return, derive a new off-policy temporal difference learning algorithm (TD-MRetrace) with linear function approximation, and obtain a convergence guarantee under standard assumptions in both prediction and control cases. Experimental results on counterexamples and control tasks validate the effectiveness of the proposed algorithm compared with traditional algorithms.

1 IMPORTANCE OF THE POSITIVE DEFINITE MATRIX

Positive definite (p.d.) matrix plays an important role in convergence analysis of reinforcement learning algorithms with linear function approximation. The convergence of on-policy TD(0) is established by Sutton [1988], where the key is the p.d. matrix $\mathbf{A}_{\text{on}}$ based on the invariance of the on-policy state distribution. Off-policy learning seeks to learn a target policy while exploring actions according to a behavior policy to avoid getting stuck in local optima. However, due to the inconsistency between the behavior policy μ and the target policy π, off-policy learning is prone to be instable, especially when combined with function approximation and bootstrapping, known as "the deadly triad" Sutton and Barto [2018]. The fundamental reason is that the positive definiteness of the matrix $\mathbf{A}_{\text{off}}$ is not guaranteed Sutton et al. [2016].

Baird et al. [1995] proposed residual algorithms by minimizing mean squared Bellman errors to solve the residual fixed point in closed-form. The key matrix is p.d., thus ensuring the stability of the algorithms. However, residual methods require double sampling in non-deterministic environments to remove dependencies between successor states. More importantly, the residual fixed point is in most cases worse than the TD fixed point Scherrer [2010].

Stable algorithms to solve the TD fixed point mainly include two approaches. Gradient based methods guarantee the positive definiteness of the correlation matrix by constructing different objective functions. Sutton et al. [2008] proposed the first convergent off-policy temporal difference learning algorithm, gradient TD (GTD), which minimizes the norm of the expected TD update (NEU) and involves a p.d. matrix $\mathbf{A}_{\text{GTD}} = \begin{pmatrix} \sqrt{\eta}\mathbf{I} & \mathbf{A}_{\text{off}} \\ -\mathbf{A}_{\text{off}}^{\top} & 0 \end{pmatrix}$, where $\mathbf{I}$ is the identity matrix and η is the stepsize ratio of the auxiliary parameter to the learning parameter. Sutton et al. [2009] proposed GTD2 algorithm with p.d. matrix $\mathbf{A}_{\text{GTD2}} = \begin{pmatrix} \sqrt{\eta}\mathbf{C} & \mathbf{A}_{\text{off}} \\ -\mathbf{A}_{\text{off}}^{\top} & 0 \end{pmatrix}$ and TD with gradient correction (TDC) algorithm with p.d. matrix $\mathbf{A}_{\text{TDC}} = \mathbf{A}_{\text{off}}^{\top}\mathbf{C}^{-1}\mathbf{A}_{\text{off}}$, both of which minimize the mean square projected Bellman error (MSPBE), where $\mathbf{C} = \mathbb{E}[\phi\phi^{\top}]$ and ϕ is feature of state or state-action pair. Hackman [2012] proposed Hybrid TD (HTD) algorithm with a p.d. matrix $\mathbf{A}_{\text{HTD}} = \mathbf{A}_{\text{off}}^{\top}\mathbf{A}_{\text{on}}^{-1}\mathbf{A}_{\text{off}}$, which replaces $\mathbf{C}^{-1}$ in $\mathbf{A}_{\text{TDC}}$ as $\mathbf{A}_{\text{on}}^{-1}$ to accelerate the learning rate. Liu et al. [2015, 2016, 2018] proposed accelerated GTD-MP and GTD2-MP algorithm via writing the objective functions, NEU and MSPBE, in the form of a convex-concave saddle-point formulation. Zhang et al. [2021] proposed Diff-GQ1 algorithm w.r.t saddle-point formulation of GTD2 and Diff-GQ2 algorithm w.r.t two-stage gradient evaluation, both of which minimize MSPBE in the average-reward setting. Pan et al. [2017] proposed accelerated TD (ATD) algorithm based on a second order update, which is a Hessian ma-

Submitted to the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023). **To be used for reviewing only**.

Table 1: Comparisons of learning algorithms with linear function approximation.

Name	definition	update rules	p.d.
TD		$\Delta\theta_t = \alpha_t \left(r_{t+1} + (\gamma\phi_{t+1} - \phi_t)^\top \theta_t\right) \phi_t$	yes
Off-policy TD	$\rho_t = \frac{\pi(a_t\|s_t)}{\mu(a_t\|s_t)}$	$\Delta\theta_t = \alpha_t\rho_t \left(r_{t+1} + (\gamma\phi_{t+1} - \phi_t)^\top \theta_t\right) \phi_t$	no
Retrace	$c_t = \min\left(1, \frac{\pi(a_t\|s_t)}{\mu(a_t\|s_t)}\right)$	$\Delta\theta_t = \alpha_t c_t \left(r_{t+1} + (\gamma\mathbb{E}_\pi[\phi_{t+1}] - \phi_t)^\top \theta_t\right) \phi_t$	no
MRetrace	$x_t = \min_a\left\{\frac{\mu(a\|s_t)}{\pi(a\|s_t)}\right\}$	$\Delta\theta_t = \alpha_t\rho_t \left(r_{t+1} + (x_{t+1}\gamma\mathbb{E}_\pi[\phi_{t+1}] - \phi_t)^\top \theta_t\right) \phi_t$	yes
TD-MRetrace		$\Delta\omega_t = \alpha_t\big[r_{t+1} + \gamma{\theta_t}^\top(\mathbb{E}_\pi[\phi_{t+1}] - \phi_{t+1}) + \gamma\omega_t^\top\phi_{t+1} - \omega_t^\top\phi_t\big]\phi_t$	yes

trix derived from MSPBE and approximated in a low rank. The main disadvantage of gradient based methods is that they need to update an additional parameter to correct the gradient and tend to converge slowly Hallak and Mannor [2017].

The other approach, importance sampling (IS) ratios, correct the returns via reweighting the state distribution between on-policy and off-policy updates. It was first proposed by Precup et al. [2001] where the p.d. matrix is $\mathbf{A}_{\text{on}}$. Sutton et al. [2016] proposed emphatic TD (ETD) algorithm with followon trace to correct from beginning of the excursion based on IS ratios, where p.d. matrix is $\mathbf{A}_{\text{ETD}} = \Phi^\top\mathbf{D}_f(\mathbf{I} - \gamma\mathbf{P}_\pi)\Phi$, $\mathbf{D}_f$ is a diagonal matrix with diagonal element approximated to $f = (\mathbf{I} - \gamma\mathbf{P}_\pi)^{-1}d_\mu$. Zhang et al. [2020] proposed convergent off-policy actor-critic algorithm in which the followon trace's variance is reduced by emphasis approximation. Zhang and Whiteson [2021] proposed truncated emphatic TD (TETD), where the p.d. matrix is $\mathbf{A}_{\text{TETD}} = \Phi^\top\mathbf{D}_{f_k}(\mathbf{I} - \gamma\mathbf{P}_\pi)\Phi$, f_k is a truncated followon trace of length k. The main disadvantage of ETD and TETD is that the followon trace may be of very high variance. Munos et al. [2016] proposed Retrace algorithm with a safe and efficient IS ratios truncated at 1, which guarantees convergence with a contraction mapping in the case of look-up table. However, based on an action-value extension to Baird's counterexample, Retrace was pointed out that it is not guaranteed to be stable when combined with function approximation Touati et al. [2018]. Then, a convergent gradient-based Retrace (GRetrace) was proposed based on a quadratic convex-concave saddle-point formulation, which minimizes MSPBE Touati et al. [2018]. However, this returns to the disadvantage of slow convergence of the gradient TD learning families.

Our contributions: In this paper, we explore modified Retrace to correct the off-policy return, and derive a new off-policy temporal difference learning algorithm (TD-MRetrace). its key matrix is positive definite, thus ensuring the learning stability.

The rest of this paper is organized as follows. First, related notations and background are introduced. Second, we revisit the fundamental reason why Retrace with linear function approximation is not stable, propose Modified Retrace (MRetrace) to correct off-policy update, and derive a off-policy learning algorithm, TD-MRetrace (see Table 1). After that, we show a convergence guarantee for TD-MRetrace algorithm under standard conditions in the off-policy setting. Finally, we experimentally verify the proposed algorithm on both prediction tasks and control tasks.

2 NOTATION AND BACKGROUND

Reinforcement learning agent interacts with its environment which we modeled as a discounted Markov Decision Process $\langle S, A, R, T, \gamma\rangle$, where S is a finite state space, $|S| = n$, A is an action space, $T : S \times A \times S \to [0, 1]$ is a transition function, $R : S \times A \times S \to \mathbb{R}$ is a reward function, $\gamma \in [0, 1)$ is a discount factor. Policy $\pi : S \times A \to [0, 1]$ offers the probability $\pi(a|s)$ to choose action a in state s. State value function for policy π, denoted $V^\pi : S \to \mathbb{R}$, represents the expected sum of discounted rewards in the MDP under policy π: $V^\pi(s) = \mathbb{E}_\pi\left[\sum_{t=0}^\infty \gamma^t r_t | s_0 = s\right]$. Action value function $Q^\pi : S \times A \to \mathbb{R}$ is defined as $Q^\pi(s, a) = \mathbb{E}_\pi\left[\sum_{t=0}^\infty \gamma^t r_t | s_0 = s, a_0 = a\right]$. V^π is the fixed point of the Bellman operator over the value function $\mathcal{T}^\pi V = r + \gamma\mathbf{P}_\pi V$, where r is the expected immediate reward and $\mathbf{P}_\pi$ denotes the $n \times n$ matrix of transition probabilities

$$[\mathbf{P}_\pi]_{ij} \doteq \sum_{a\in A} \pi(a|i)T(i, a, j). \tag{1}$$

Assume the state distribution d_π under policy π is steady and exists. Then one special property is the invariance of distribution d_π,

$$d_\pi = \mathbf{P}_\pi^\top d_\pi. \tag{2}$$

When the state space is too large to preserve $V^\pi(s)$, a linear function approximation is used to generalize between different states $V^\pi(s) \approx V_\theta(s) = \theta^\top\phi(s) = \sum_{i=1}^m \theta_i\phi_i(s)$, where θ is the weight vector, $\phi(s)$ is the feature vector of state s, and the feature size is far less than the state space $m \ll n$. The action value function is generalized as $Q(s, a) \approx Q_\theta(s, a) = \theta^\top\phi(s, a)$, where $\phi(s, a)$ is the feature vector of state action pair. Notably, equation $V_\theta = \mathcal{T}^\pi V_\theta$ no longer holds because the number of parameters is far less than the number of equations. A common and efficient solution is the TD fixed point $V_\theta = \Pi\mathcal{T}^\pi V_\theta$

with projection $\Pi = \Phi(\Phi^\top \mathbf{D}_\pi \Phi)^{-1}\Phi^\top \mathbf{D}_\pi$, where Φ is the $n \times m$ matrix with the $\phi(s)$ as its rows, $\mathbf{D}_\pi$ is the $n \times n$ diagonal matrix with d_π on its diagonal. It can be learned by the on-policy TD(0) algorithm:

$$\begin{aligned}\theta_{t+1} \doteq & \theta_t + \alpha_t \left(r_{t+1} + \gamma \theta_t^\top \phi_{t+1} - \theta_t^\top \phi_t\right) \phi_t \\ = & \theta_t + \alpha_t \left(r_{t+1}\phi_t - \phi_t(\phi_t - \gamma\phi_{t+1})^\top \theta_t\right),\end{aligned} \tag{3}$$

where $\alpha_t > 0$ is a step-size parameter, and we have used the shorthand $\phi_t \doteq \phi(s_t)$. Algorithms' convergence analysis with linear function approximation is mainly based on the ODE (Ordinary Differential Equations) approach Borkar and Meyn [2000], where the key relies on the matrix $\mathbf{A}$ being positive definite, i.e. $\forall x \neq 0$, $x^\top \mathbf{A} x > 0$. Let $\mathbf{A}_{\text{on}}$ denote the key matrix of the expected update (3):

$$\begin{aligned}\mathbf{A}_{\text{on}} &= \lim_{t\to\infty} \mathbb{E}_\pi \left[\phi_t(\phi_t - \gamma\phi_{t+1})^\top\right] \\ &= \Phi^\top \mathbf{D}_\pi (\mathbf{I} - \gamma \mathbf{P}_\pi)\Phi.\end{aligned} \tag{4}$$

With property (2), $\mathbf{A}_{\text{on}}$ is proved to be p.d., thus the convergence of the on-policy TD algorithm is established Sutton [1988].

In this paper, we are concerned with off-policy learning, where the target policy π is different from the behavior policy μ that generates experiences $\langle s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}\rangle$. There are two ways to implement the off-policy learning. One is to use the experiences of the behavior policy and simply multiplies the whole on-policy TD update (3) by the importance sampling ratio $\rho_t = \frac{\pi(a_t|s_t)}{\mu(a_t|s_t)}$, e.g., off-policy TD:

$$\begin{aligned}\theta_{t+1} \doteq & \theta_t + \rho_t \alpha_t (r_{t+1} + \gamma \theta_t^\top \phi_{t+1} - \theta_t^\top \phi_t)\phi_t \\ = & \theta_t + \alpha_t \left(\rho_t r_{t+1}\phi_t - \rho_t \phi_t(\phi_t - \gamma\phi_{t+1})^\top \theta_t\right).\end{aligned} \tag{5}$$

Its key matrix is:

$$\begin{aligned}\mathbf{A}_{\text{off}} &= \lim_{t\to\infty} \mathbb{E}_\mu \left[\rho_t \phi_t(\phi_t - \gamma\phi_{t+1})^\top\right] \\ &= \lim_{t\to\infty} \mathbb{E}_\mu \left[\frac{\pi(a|s)}{\mu(a|s)}\phi_t(\phi_t - \gamma\phi_{t+1})^\top\right] \\ &= \lim_{t\to\infty} \mathbb{E}_\pi \left[\phi_t(\phi_t - \gamma\phi_{t+1})^\top\right] \\ &= \Phi^\top \mathbf{D}_\mu (\mathbf{I} - \gamma\mathbf{P}_\pi)\Phi,\end{aligned} \tag{6}$$

The other is to directly use the target policy:

$$\begin{aligned}\theta_{t+1} \doteq & \theta_t + \alpha_t (r_{t+1} + \gamma \theta_t^\top \mathbb{E}_\pi[\phi_{t+1}] - \theta_t^\top \phi_t)\phi_t \\ = & \theta_t + \alpha_t \left(r_{t+1}\phi_t - \phi_t(\phi_t - \gamma\mathbb{E}_\pi[\phi_{t+1}])^\top \theta_t\right).\end{aligned} \tag{7}$$

Q-learning can be seen as a special case of (7), if we set $\theta_t^\top \mathbb{E}_\pi[\phi_{t+1}] \doteq \max_a \theta_t^\top \phi(s_{t+1}, a) = \max_a Q_{\theta_t}(s_{t+1}, a)$. The key matrix of these two off-policy learning algorithms share the same form $\mathbf{A}_{\text{off}} = \Phi^\top \mathbf{D}_\mu (\mathbf{I} - \gamma\mathbf{P}_\pi)\Phi$.

2.1 2-STATE COUNTEREXAMPLE

The $\theta \to 2\theta$ problem has only two states Tsitsiklis and Van Roy [1997], Sutton et al. [2016]. 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 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 of the behavior policy is $d_\mu = (0.5, 0.5)^\top$. The discount factor is $\gamma = 0.9$.

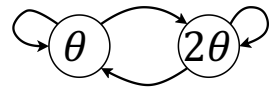


Figure 1: The 2-state counterexample.

For the counterexample, the key matrix of the off-policy TD is $\mathbf{A}_{\text{off}} = \Phi^\top \mathbf{D}_\mu (\mathbf{I} - \gamma\mathbf{P}_\pi)\Phi = -0.2$. This means that off-policy TD is not stable.

2.2 INSTABILITY OF RETRACE

Retrace algorithm belongs to the second implementation of off-policy learning. It employs a truncated IS ratios $c_t = \min(1, \rho_t)$ and guarantees convergence with look-up value function Munos et al. [2016]. We revisit Retrace(0) with linear function approximation by Touati et al. [2018], where the truncated IS ratios are multiplied to the whole TD error:

$$\begin{aligned}\theta_{t+1} \doteq & \theta_t + c_t \alpha_t \left(r_{t+1} + \gamma \theta_t^\top \mathbb{E}_\pi[\phi_{t+1}] - \theta_t^\top \phi_t\right) \phi_t \\ = & \theta_t + \alpha_t \left(c_t r_{t+1}\phi_t - c_t \phi_t(\phi_t - \gamma\mathbb{E}_\pi[\phi_{t+1}])^\top \theta_t\right),\end{aligned} \tag{8}$$

where $\mathbb{E}_\pi[\phi_{t+1}] = \sum_a \pi(a|s_{t+1})\phi(s_{t+1})$. The key matrix of the expected Retrace's update (8) is:

$$\begin{aligned}\mathbf{A}_{\text{Retrace(0)}} &= \lim_{t\to\infty} \mathbb{E}_\mu \left[c_t \phi_t(\phi_t - \gamma\mathbb{E}_\pi[\phi_{t+1}])^\top\right] \\ &= \Phi^\top \mathbf{D}_\mu \mathbf{D}_c (\mathbf{I} - \gamma\mathbf{P}_\pi)\Phi,\end{aligned} \tag{9}$$

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

$$d_c(s) = \sum_a \min(\mu(a|s), \pi(a|s)). \tag{10}$$

In the counterexample, according to (10), $d_c = (0.5, 0.5)^\top$. Then, the key matrix of Retrace(0) algorithm for this example is: $\mathbf{A}_{\text{Retrace(0)}} = \Phi^\top \mathbf{D}_\mu \mathbf{D}_c (\mathbf{I} - \gamma\mathbf{P}_\pi)\Phi = -0.1$. Thus, Retrace(0) with linear function approximation is not stable.

3 TD-MRETRACE ALGORITHM

In this section we propose a mechanism to correct off-policy update and derive new algorithms.

3.1 MODIFIED RETRACE

Importance sampling ratios, $\rho_t = \frac{\pi(a_t|s_t)}{\mu(a_t|s_t)}$, represent the "off-policyness" of the current state and action between the target policy and the behavior policy. The farther the target policy deviates, the more unstable the learning algorithm will be. In this sense, the maximum of the "off-policyness", $\max_a \rho_t = \max_a \{\frac{\pi(a_t|s_t)}{\mu(a_t|s_t)}\}$, is the key to the instability of off-policy learning algorithms.

In order to reduce the impact of the deviation of the target policy, we introduce modified retrace (MRetrace) that takes the reciprocal of the above maximum degree as follows:

$$x(s_t) \doteq \frac{1}{\max_a \rho_t} = \min_a \Big\{\frac{1}{\rho_t}\Big\} = \min_a \Big\{\frac{\mu(a|s_t)}{\pi(a|s_t)}\Big\}. \quad (11)$$

Obviously, $x(s_t) \leq 1$[1], and $x(s_t) = 1$ only when $\forall a$, $\pi(a|s_t) = \mu(a|s_t)$.

3.1.1 MRetrace learning for prediction

We use the first way to learn state values for prediction. The resulting temporal difference learning algorithm, which we call MRetrace learning, is

$$\begin{aligned}\theta_{t+1} \doteq & \theta_t + \alpha_t \rho_t \big(r_{t+1} + x_t \gamma \theta_t^\top \phi_{t+1} - \theta_t^\top \phi_t\big)\phi_t \\ = & \theta_t + \alpha_t \big(\rho_t r_{t+1} \phi_t - \rho_t \phi_t (\phi_t - x_t \gamma \phi_{t+1})^\top \theta_t\big) \\ = & \theta_t + \alpha_t \big(\mathbf{b}_t - \mathbf{A}_t \theta_t\big),\end{aligned} \quad (12)$$

where x_t is in short of $x(s_t)$, $\mathbf{b}_t = \rho_t r_{t+1} \phi_t$, $\mathbf{A}_t = \rho_t \phi_t (\phi_t - x_t \gamma \phi_{t+1})^\top$. Then,

$$\mathbf{b} = \lim_{t\to\infty} \mathbb{E}_\mu[\mathbf{b}_t] = \lim_{t\to\infty} \mathbb{E}_\mu [\rho_t r_{t+1} \phi_t] = \Phi^\top \mathbf{D}_\mu r_\pi, \quad (13)$$

where r_π is expected reward vector under policy π with each component $r_\pi(s) = \sum_a \sum_{s'} \pi(a|s) R(s,a,s')$. The key matrix of MRetrace is

$$\begin{aligned}\mathbf{A} = \lim_{t\to\infty} \mathbb{E}_\mu [\mathbf{A}_t] &= \lim_{t\to\infty} \mathbb{E}_\mu \left[\rho_t \phi_t (\phi_t - x_t \gamma \phi_{t+1})^\top\right] \\ &= \lim_{t\to\infty} \mathbb{E}_\mu \left[\frac{\pi(a|s)}{\mu(a|s)} \phi_t (\phi_t - x_t \gamma \phi_{t+1})^\top\right] \\ &= \lim_{t\to\infty} \mathbb{E}_\pi \left[\phi_t (\phi_t - x_t \gamma \phi_{t+1})^\top\right] \\ &= \Phi^\top \mathbf{D}_\mu (\mathbf{I} - \gamma \mathbf{D}_x \mathbf{P}_\pi) \Phi,\end{aligned} \quad (14)$$

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

[1]Note that $\sum_a \mu(a|s_t) = 1$, $\sum_a \pi(a|s_t) = 1$.

3.1.2 MRetrace learning for control

We use the second way to learn action values for control. The update rule is as follows:

$$\begin{aligned}\theta_{t+1} &= \theta_t + \alpha_t \rho_t (r_{t+1} + x_{t+1} \gamma \theta_t^\top \mathbb{E}_\pi[\phi_{t+1}] - \theta_t^\top \phi_t)\phi_t \\ &= \theta_t + \alpha_t \rho_t (r_{t+1} \phi_t - \phi_t (\phi_t - x_{t+1} \gamma \mathbb{E}_\pi[\phi_{t+1}])^\top \theta_t) \\ &= \theta_t + \alpha_t (\mathbf{b}_t - \mathbf{A}_t \theta_t),\end{aligned} \quad (15)$$

where $\mathbf{b}_t = \rho_t r_{t+1} \phi_t$, and $\mathbf{A}_t = \rho_t \phi_t (\phi_t - x_{t+1} \gamma \mathbb{E}_\pi[\phi_{t+1}])^\top$. Then,

$$\mathbf{b} = \lim_{t\to\infty} \mathbb{E}_\mu[\mathbf{b}_t] = \lim_{t\to\infty} \mathbb{E}_\mu[\rho_t r_{t+1} \phi_t] = \Phi^\top \mathbf{D}_\mu r_\pi. \quad (16)$$

The key matrix is

$$\begin{aligned}\mathbf{A} = \lim_{t\to\infty} \mathbb{E}_\mu[\mathbf{A}_t] &= \lim_{t\to\infty} \mathbb{E}_\mu[\rho_t \phi_t (\phi_t - x_{t+1} \gamma \mathbb{E}_\pi[\phi_{t+1}])^\top] \\ &= \Phi^\top \mathbf{D}_\mu (\mathbf{I} - \gamma \mathbf{D}_x \mathbf{P}_\pi) \Phi.\end{aligned} \quad (17)$$

It is worth noting that if we remove the important sampling ratios ρ_t from (15), the key matrix $\mathbf{A}$ remains the same since the terms, r_{t+1}, ϕ_t and $\mathbb{E}_\pi[\phi_{t+1}]$ in the state action values are independent of ρ_t. But we still keep ρ_t for reasons explained below. When the successor state is composed of afterstate and dynamics, e.g., Tetris game, one usually learn the afterstate values. The distribution of these afterstates is generated by the behavior policy. Therefore, ρ_t is needed to correct the target returns.

From (13), (14), (16) and (17), we can see that the expectation of updates for MRetrace learning algorithms share the same form. The only difference is that the feature matrix is defined on state for prediction and on state-action pair for control.

For the 2-state counterexample, $d_x = (0.5, 0.5)^\top$, the value of the new key matrix is as follows:

$$\mathbf{A} = \Phi^\top \mathbf{D}_\mu (\mathbf{I} - \gamma \mathbf{D}_x \mathbf{P}_\pi) \Phi = 1.15. \quad (18)$$

This shows that our algorithm is convergent on this counterexample. The specific theoretical proof and experimental verification are left to following sections.

3.2 THE SOLVED TD FIXED POINT

MRetrace enhances stability by reducing the impact of off-policy. It is important to show what solution it seeks.

When the parameter vector θ in (12) is no longer updated, it means that the MRetrace algorithm converges. In this case, $\mathbf{b} - \mathbf{A}\theta = 0$. That is $\theta = \mathbf{A}^{-1}\mathbf{b}$ if $\mathbf{A}$ is reversible. It is the solution to the following expectation equation:

$$\mathbb{E}_\mu \left[\rho(r + x\gamma\theta^\top \mathbb{E}_\pi[\phi'] - \theta^\top \phi)\phi\right] = 0. \quad (19)$$

Lemma 3.1. *The TD fixed point (19) follows from $V_\theta = \Pi \mathcal{T}_x^\pi V_\theta$, where the modified Bellman operator $\mathcal{T}_x^\pi$ is defined as*

$$\mathcal{T}_x^\pi V \doteq r + \gamma \boldsymbol{D}_x \boldsymbol{P}_\pi V. \tag{20}$$

Proof.

$$\begin{aligned} 0 &= \mathbb{E}_\mu \left[\rho(r + x\gamma\theta^\top \mathbb{E}_\pi[\phi'] - \theta^\top \phi)\phi\right] \\ &= \sum_s d_s \mathbb{E}_\pi[(r + x\gamma V_\theta(s') - V_\theta(s))\phi(s)] \\ &= \Phi^\top \mathbf{D}_\mu (\mathcal{T}_x^\pi V_\theta - V_\theta). \end{aligned} \tag{21}$$

We have

$$\begin{aligned} \Phi^\top \mathbf{D}_\mu \mathcal{T}_x^\pi V_\theta &= \Phi^\top \mathbf{D}_\mu V_\theta \\ &= \Phi^\top \mathbf{D}_\mu \Phi \theta. \end{aligned} \tag{22}$$

Then, $\theta = (\Phi^\top \mathbf{D}_\mu \Phi)^{-1} \Phi^\top \mathbf{D}_\mu \mathcal{T}_x^\pi V_\theta$. That is $V_\theta = \Phi\theta = \Phi(\Phi^\top \mathbf{D}_\mu \Phi)^{-1} \Phi^\top \mathbf{D}_\mu \mathcal{T}_x^\pi V_\theta = \Pi \mathcal{T}_x^\pi V_\theta$. □

According to Scherrer [2010], (19) is TD fixed point due to the projection direction ($\mathbf{D}_\mu \Phi$) in the projection operator Π. Note that it is neither the TD fixed point of the behavior policy, nor the exact TD fixed point of the target policy in MDP $\langle S, A, R, T, \gamma \rangle$.

Define a discount variable $\gamma^{\mu,\pi}$ on state s as $\gamma^{\mu,\pi}(s) = \gamma x(s)$. Then, the modified Bellman operator $\mathcal{T}_x^\pi$ in MDP $\langle S, A, R, T, \gamma \rangle$ equals to the Bellman operator $\mathcal{T}^\pi$ in MDP $\langle S, A, R, T, \gamma^{\mu,\pi} \rangle$.

Thus, MRetrace (12) solves the TD fixed point of the target policy in MDP $\langle S, A, R, T, \gamma^{\mu,\pi} \rangle$.

3.3 TD-MRETRACE ALGORITHM

Remember that our objective is to solve the TD fixed point of the target policy in MDP $\langle S, A, R, T, \gamma \rangle$.

Consider another weight vector ω, the off-policy TD error $\delta^\pi(\omega_t)$ can be decomposed as follows:

$$\begin{aligned} \delta^\pi(\omega_t) &= r_{t+1} + \gamma \omega_t^\top \mathbb{E}_\pi[\phi_{t+1}] - \omega_t^\top \phi_t \\ &= r_{t+1} + \gamma \omega_t^\top (\mathbb{E}_\pi[\phi_{t+1}] - \phi_{t+1} + \phi_{t+1}) - \omega_t^\top \phi_t \\ &= r_{t+1} + \gamma \omega_t^\top (\phi_{t+1} - \phi_t) + \gamma \omega_t^\top (\mathbb{E}_\pi[\phi_{t+1}] - \phi_{t+1}) \\ &= \delta^\mu(\omega_t) + \delta^{\text{off}}(\omega_t), \end{aligned} \tag{23}$$

where the on-policy TD error $\delta^\mu(\omega_t) \doteq r_{t+1} + \gamma \omega_t^\top (\phi_{t+1} - \phi_t)$, the off-policy correction $\delta^{\text{off}}(\omega_t) \doteq \gamma \omega_t^\top (\mathbb{E}_\pi[\phi_{t+1}] - \phi_{t+1})$.

It is a hybrid approach that combines the on-policy update and the off-policy update together Hackman [2012]. When our target and behavior policy are the same, $\mathbb{E}_\pi[\phi_{t+1}] - \phi_{t+1} = 0$, the update becomes the expected Sarsa update. Therefore, the instability is due to the off-policy correction $\delta^{\text{off}}(\omega_t)$.

Let the off-policy correction be approximated as $\delta^{\text{off}}(\omega_t) \approx \delta^{\text{off}}(\theta_t)$ based on the proposed MRetrace. Then, the off-policy TD error can be approximated as follows:

$$\begin{aligned} \delta^\pi(\omega_t) &= \delta^\mu(\omega_t) + \delta^{\text{off}}(\omega_t) \\ &\approx \delta^\mu(\omega_t) + \delta^{\text{off}}(\theta_t) \\ &= r_{t+1} + \gamma \omega_t^\top (\phi_{t+1} - \phi_t) + \gamma \theta_t^\top (\mathbb{E}_\pi[\phi_{t+1}] - \phi_{t+1}) \end{aligned} \tag{24}$$

The resultant algorithm, which we call TD-MRetrace, is

$$\begin{aligned} \omega_{t+1} =& \omega_t + \alpha_t \delta^\mu(\omega_t)\phi_t + \alpha_t \delta^{\text{off}}(\theta_t)\phi_t \\ =& \omega_t + \alpha_t \left[r_{t+1} + \gamma \omega_t^\top (\phi_{t+1} - \phi_t)\right]\phi_t \\ &+ \alpha_t \gamma {\theta_t}^\top (\mathbb{E}_\pi[\phi_{t+1}] - \phi_{t+1})\phi_t. \end{aligned} \tag{25}$$

where θ_t is generated by (12). Note that the update to ω_t is the sum of two terms, and that the first term is exactly the same as the on-policy update. The second term has nothing to do with ω and can be regarded as a correction of the reward in the off-policy case. Once θ converges, ω will converges such as on-policy TD learning.

4 CONVERGENCE

The purpose of this section is to establish that the TD-MRetrace algorithm converges with probability one under standard assumptions when $\{\phi_t, r_t, \mathbb{E}_\pi[\phi_{t+1}]\}$ is obtained by the off-policy subsampling process Sutton et al. [2008].

Let s be a state randomly drawn from d_μ, and let s' be a state obtained by following π for one time step in the MDP from s. Let the behavior policy μ select all actions of the target policy π with positive probability in every state, and the target policy is deterministic. Further, let $r(s, s')$ be the reward incurred.

Assumption 4.1. *The Markov chain (s_t) is aperiodic and irreducible, so that* $\lim_{t\to\infty} \mathbb{P}(s_t = s' | s_0 = s) = d_\mu(s')$ *exists and is unique.*

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

$$d_\mu = \mathbf{P}_\mu^\top d_\mu, \tag{26}$$

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

$$[\mathbf{P}_\mu]_{ss'} = \sum \mu(a|s) T(s, a, s'). \tag{27}$$

Assumption 4.2. $\{\phi_t, r_t, \mathbb{E}_\pi[\phi_{t+1}]\}$ *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 4.3. *The feature matrix Φ is column full rank.*

Assumption 4.4. *Step-size sequence α_t satisfies $\alpha_t \in (0, 1]$, $\sum_{t=0}^{\infty} \alpha_t = \infty$, and $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$.*

Theorem 4.5. *(Convergence of MRetrace with an off-policy sub-sampled process). Assume Assumption 4.1, 4.2, 4.3, and 4.4. Let the parameter θ_t be updated by iteration (12). Let $\boldsymbol{A} = \mathbb{E}_\mu \left[\rho_t \phi_t (\phi_t - x_t \gamma \mathbb{E}_\pi[\phi_{t+1}])^\top\right]$, $\boldsymbol{b} = \mathbb{E}_\mu[\rho_t r_t \phi_t]$. Then the parameter vector θ_t converges with probability one to the TD fixed-point $\theta^* = \boldsymbol{A}^{-1}\boldsymbol{b}$ (19).*

Proof. The proof follows from the procedures of Sutton et al. [2008, 2009] for GTD and GTD2, which are based on the ordinary-differential-equation (ODE) approach Borkar and Meyn [2000]. First, $\mathbf{A}$ and $\mathbf{b}$ are well-defined according to Assumption 4.1 and 4.2.

Now we apply Theorem 2.2 of Borkar and Meyn [2000]. We write $\theta_{t+1} = \theta_t + \alpha_t(-\mathbf{A}\theta_t + \mathbf{b} + (\mathbf{A} - \mathbf{A}_{t+1})\theta_t + (\mathbf{b}_{t+1} - \mathbf{b})) = \theta_t + \alpha_t(h(\theta_t) + M_{t+1})$, where $h(\theta) = \mathbf{b} - \mathbf{A}\theta$ and $M_{t+1} = (\mathbf{A} - \mathbf{A}_{t+1})\theta_t + \mathbf{b}_{t+1} - \mathbf{b}$. Let $\mathcal{F}_t = \sigma(\theta_1, M_1, \ldots, \theta_{t-1}, M_t)$. Theorem 2.2 requires the verification of the following conditions: (i) The function h is Lipschitz and $h_\infty(\theta) = \lim_{r\to\infty} h(r\theta)/r$ is well-defined for every $\theta \in \mathbb{R}^m$; (ii-a) The sequence $(M_t, \mathcal{F}_t)$ is a martingale difference sequence, and (ii-b) for some $C_0 > 0$, $\mathbb{E}[||M_{t+1}||^2|\mathcal{F}_t] \le C_0(1 + ||\theta_t||^2)$ holds for any initial parameter vector θ_1; (iii) The sequence α_t satisfies $0 < \alpha_t \le 1$, $\sum_{t=0}^{\infty} \alpha_t = \infty$, and $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$; (iv) The ODE $\dot{\theta} = h_\infty(\theta)$ has the origin as a globally asymptotically stable equilibrium; and (v) The ODE $\dot{\theta} = h(\theta)$ has a unique globally asymptotically stable equilibrium.

Clearly, $h(\theta)$ is Lipschitz with coefficient $||\mathbf{A}||$ and $h_\infty(\theta) = -\mathbf{A}\theta$. By construction, $(M_t, \mathcal{F}_t)$ satisfies $\mathbb{E}[M_{t+1}|\mathcal{F}_t] = 0$ and $M_t \in \mathcal{F}_t$, i.e., it is a martingale difference sequence. Condition (ii-b) can be shown to hold by a simple application of the triangle inequality and the boundedness of the second moments of $\{\phi_t, r_t, \phi'_t\}_t$. Condition (iii) is satisfied by our conditions on the step-size sequences α_t.

For the last two conditions, we begin by showing that the matrix $\mathbf{A} = \mathbb{E}_\mu[\phi_t(\phi_t - x_t\gamma\mathbb{E}_\pi[\phi_{t+1}])^\top] = \Phi^\top \mathbf{D}_\mu(\mathbf{I} - \gamma\mathbf{D}_x\mathbf{P}_\pi)\Phi$ is p.d. Note that $\mathbf{A}$ consists of $\Phi^\top$ and Φ wrapped around an $n \times n$ matrix $\mathbf{D}_\mu(\mathbf{I} - \gamma\mathbf{D}_x\mathbf{P}_\pi)$. According to assumption that the feature matrix Φ is column full rank, then, $\mathbf{A}$ is p.d. whenever the key matrix $\mathbf{D}_\mu(\mathbf{I} - \gamma\mathbf{D}_x\mathbf{P}_\pi)$ is p.d.

Based on two theorems showed by Sutton [1988], Sutton et al. [2016], positive definiteness of the key matrix is assured if all of its columns and rows sum to positive numbers. One theorem is that any matrix $\mathbf{M}$ is p.d. if and only if the symmetric matrix $\mathbf{S} = \mathbf{M} + \mathbf{M}^\top$ is p.d. Another theorem is that any symmetric real matrix $\mathbf{S}$ is p.d. if the absolute values of its diagonal entries are greater than the sum of the absolute values of the corresponding off-diagonal entries. For the key matrix, $\mathbf{M} = \mathbf{D}_\mu(\mathbf{I} - \gamma\mathbf{D}_x\mathbf{P}_\pi)$, the diagonal entries are positive and the off-diagonal entries are negative, so all we have to show is that all components of both ($\mathbf{M1}$) and ($\mathbf{1}^\top\mathbf{M}$) are positive, where $\mathbf{1}$ is the column vector with all components equal to 1. They can be verified as follows:

$$\begin{aligned}\mathbf{M1} = \mathbf{D}_\mu(\mathbf{I} - \gamma\mathbf{D}_x\mathbf{P}_\pi)\mathbf{1} &= \mathbf{D}_\mu(\mathbf{1} - \gamma\mathbf{D}_x\mathbf{P}_\pi\mathbf{1}) \\ &= \mathbf{D}_\mu(\mathbf{1} - \gamma\mathbf{D}_x\mathbf{1}) \\ &= \mathbf{D}_\mu(\mathbf{1} - \gamma d_x)\end{aligned} \tag{28}$$

Each component of $\mathbf{M1}$ is $[\mathbf{D}_\mu(\mathbf{1} - \gamma d_x)](s) = d_\mu(s)(1 - \gamma \min_b\{\frac{\mu(b|s)}{\pi(b|s)}\}) \ge d_\mu(s)(1-\gamma) > 0$.

$$\begin{aligned}[\mathbf{D}_x\mathbf{P}_\pi]_{ij} &= \min_b\Big\{\frac{\mu(b|i)}{\pi(b|i)}\Big\}\sum_a \pi(a|i)T(i,a,j) \\ &= \sum_a \pi(a|i)\min_b\Big\{\frac{\mu(b|i)}{\pi(b|i)}\Big\}T(i,a,j) \\ &\le \sum_a \pi(a|i)\frac{\mu(a|i)}{\pi(a|i)}T(i,a,j) \\ &= \sum_a \mu(a|i)T(i,a,j) \\ &= [\mathbf{P}_\mu]_{ij}.\end{aligned} \tag{29}$$

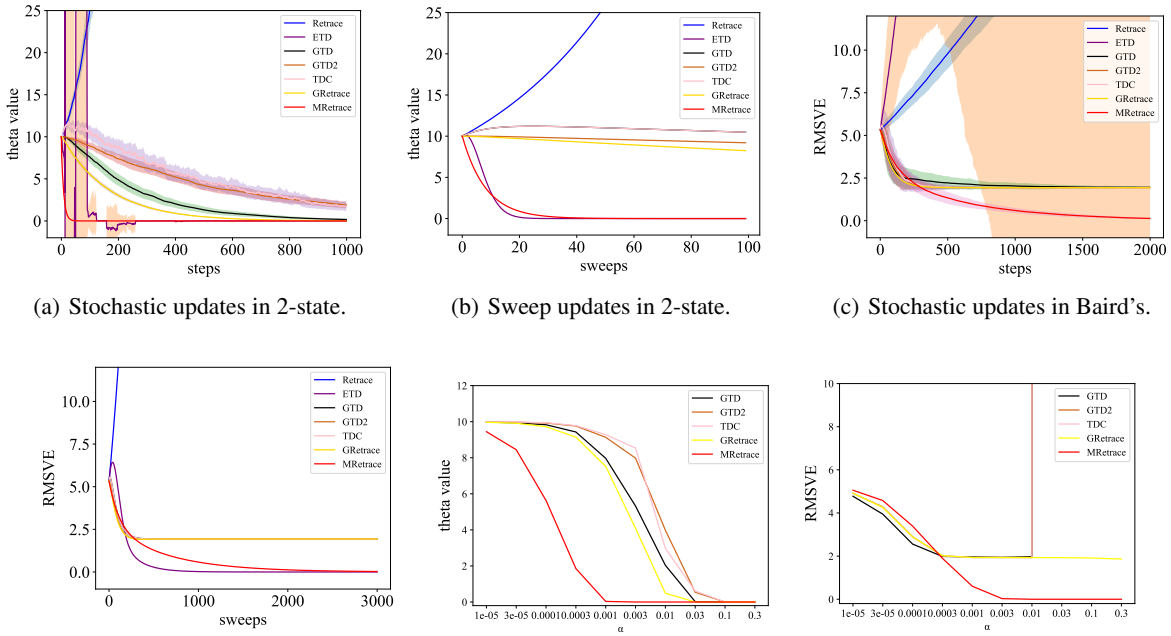
$$\begin{aligned}\mathbf{1}^\top\mathbf{M} &= \mathbf{1}^\top\mathbf{D}_\mu(\mathbf{I} - \gamma\mathbf{D}_x\mathbf{P}_\pi) \\ &= d_\mu^\top(\mathbf{I} - \gamma\mathbf{D}_x\mathbf{P}_\pi) \\ &= d_\mu^\top - \gamma d_\mu^\top\mathbf{D}_x\mathbf{P}_\pi \\ &\ge d_\mu^\top - \gamma d_\mu^\top\mathbf{P}_\mu \\ &= d_\mu^\top - \gamma d_\mu^\top \\ &= (1-\gamma)d_\mu^\top\end{aligned} \tag{30}$$

Each component of the vector $\mathbf{1}^\top\mathbf{M}$ is $[(1-\gamma)d_\mu](s) = (1-\gamma)d_\mu(s) > 0$. The row sums and the column sums are all positive. Thus, (iv) is satisfied.

Finally, for the ODE $\dot{\theta} = h(\theta)$, note that $\theta^* = A^{-1}b$ is the unique asymptotically stable equilibrium with $\overline{V}(\theta) = \frac{1}{2}||-A\theta + b||^2$ as its associated strict Liapunov function. The claim now follows. □

Theorem 4.6. *(Convergence of TD-MRetrace with an off-policy sub-sampled process). Assume Assumption 4.1, 4.2, 4.3, and 4.4. Let the parameter ω_t be updated by iteration (25) and θ_t be updated by iteration (12). Then the parameter vector ω_t converges with probability one.*

Proof. A sketch proof is given as follows. Based on Theorem 4.5, θ converges. Then, $\delta^{\text{off}}(\theta_t)$ is stable. Let a new reward $r_{t+1}^{\text{new}} \doteq \delta^{\text{off}}(\theta_t) + r_{t+1}$, this reward can be regarded as a correction to reward function in the off-policy case. Therefore, TD-MRetrace is actually an on-policy TD learning algorithm. It is guaranteed to converge, just like TD. □



(a) Stochastic updates in 2-state. (b) Sweep updates in 2-state. (c) Stochastic updates in Baird's.

(d) Sweep updates in Baird's. (e) Sensitive test in 2-state (f) Sensitive test Baird's

Figure 2: Comparisons of various temporal difference updates in counterexamples.

5 EXPERIMENTAL STUDIES

In experiments, we care about two points about the proposed TD-MRetrace algorithm: (1) Whether it converges experimentally, although it does converge in theory? (2) What is the quality of the TD fixed point it solves? We adopted two sets of experiments, i.e., counterexamples to test the stability and control tasks to test the utility.

5.1 ABOUT STABILITY IN COUNTEREXAMPLES

In the 2-states counterexample and Baird's counterexample, we implemented two update styles including stochastic updates and sweep updates, and finished parameter sensitivity test for converged algorithms. Compared algorithms include Retrace, ETD, GTD, GTD2, TDC, and GRetrace. Each algorithm was run 100 times independently.

Algorithms' learning curves including mean in line and standard deviation in shaded regions and sensitive testing are shown in Figure 2, where the theta value is equal to the root of mean squared value error (RMSVE) since there is only one scalar parameter in the 2-state counterexample and the true value is zero. We can see that (i) Retrace diverges in all cases. (ii) Expected ETD converges to zero the fastest. On the other hand, ETD converges with a high variance at the beginning in the 2-state counterexample, and diverges in Baird's counterexample which is consistent with results of computational experiments about ETD introduced in Page 282 of Sutton and Barto [2018]. (iii) MRetrace converges to zero relatively fast in all cases. (iv) MRetrace performs best in parameter sensitivity tests.

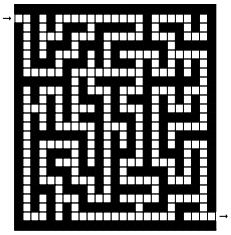
This result is not surprising. MRetrace updates only one parameter vector, and thus avoids the slow convergence problem compared to the gradient-based approaches.

5.2 LEARNING TO CONTROL

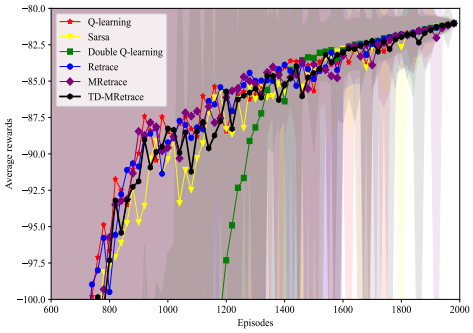
We divided into two groups of experiments to test the solution quality. In the first set of experiments, we removed function approximation from "the deadly triad", and used tabular value functions instead. Under this settings, algorithms should converge. What we care about is that whether the proposed algorithm can obtain the optimal solution. Therefore, we adopted the classic maze task. In the second set of experiments, we directly address "the deadly triad": linear function approximation, bootstrapping, and off-policy learning. Therefore, we adopted the classic Tetris task, which was used as a benchmark challenge for various optimization techniques including reinforcement learning.

5.2.1 25×25 Maze

We use a 25×25 version of Maze, see Figure 3(a). Reward for each step is set to -1, except for the end state which is 0. The action value for each state action pair is initialized to 0. The behavior policy is an ϵ-greedy policy. ϵ is initialized to



(a) Maze map



(b) Learning curves

Figure 3: Comparisons of learning algorithms in Maze.

0.1, and decreases to 0 with episodes. Compared algorithms include Q-learning, Sarsa, Retrace and Double Q-learning. Each algorithm was run 1000 times independently.

Algorithms' learning curves including mean in line and standard deviation in shaded regions are shown in Figure 3(b). We can see that (i) As expected, each algorithm converges and converges to the optimal policy since there are no "deadly triad". (ii) Double-Q learning converges the slowest because it has twice the learning parameters. (iii) Sarsa learning converges slower than Q-learning because Sarsa is not an off-policy learning. Its convergence to the optimal policy is due to the decrease of ϵ in the behavior policy. (iv) Q-learning, Retrace, MRetrace and TD-MRetrace perform well with no significant differences.

5.2.2 10×10 Tetris

Tetris game is used as a challenge for various optimization techniques Thiery and Scherrer [2009], where value function based reinforcement learning algorithms have performed extremely poorly, i.e., removing only about 50 lines on average in the 20×10 version of Tetris game where the reward is set to one point for each line removed Gabillon et al. [2013]. It is much harder to learn in the 10×10 version of Tetris. We learn the afterstate values via linear summation with weighted DT9 features Scherrer et al. [2015], which are normalized in [0,1].

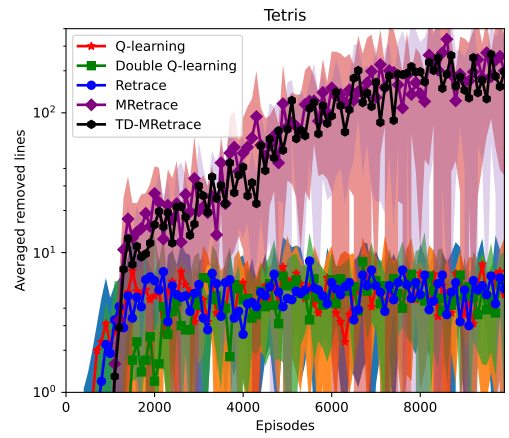


Figure 4: Comparisons of learning curves in the 10×10 tetris tasks.

For the hyperparameter settings, the learning rate α is fixed at 0.001 with no decay. The initial ϵ is set to 0.01 and decays to 0.0001 with a decay rate of 0.9992. Compared algorithms include Q-learning, Retrace and Double Q-learning. Each algorithm was run 10 times independently.

Algorithms' learning curves including mean in line and standard deviation in shaded regions are shown in Figure 4, where the averaged removed lines represent the expected return per episode. We can see that (i) On the 10×10 version of Tetris, Q-learning and Double Q-learning perform poorly but that is consistent with the literature. (ii) Although not reaching the state of the art, MRetrace and TD-MRetrace perform much better than the other three algorithms. To the best of our knowledge, MRetrace and TD-MRetrace are the first two discounted value function based reinforcement learning algorithms that perform well on Tetris.

In summary, the experiments verified the convergence of the TD-MRetrace algorithm. Moreover, in terms of quality testing, it finds a relatively good policy, although it solves an approximation of the target policy.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a simple but efficient method by introducing modified retrace to correct the return of the target policy, and guarantee the convergence of the proposed TD-MRetrace algorithm. The effectiveness of TD-MRetrace with linear value functions are validated in both evaluation tasks and control tasks.

Future work include: (i) extensions of TD-MRetrace(0) with the one-step update to TD-MRetrace(λ) with multi-step updates. (ii) extensions of the proposed TD-MRetrace algorithm with nonlinear value functions.

References

Leemon Baird et al. Residual algorithms: Reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Machine Learning*, pages 30–37, 1995.

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

Victor Gabillon, Mohammad Ghavamzadeh, and Bruno Scherrer. Approximate dynamic programming finally performs well in the game of tetris. In *Advances in Neural Information Processing Systems*, pages 1754–1762. Cambridge, MA: MIT Press, 2013.

Leah M Hackman. *Faster Gradient-TD Algorithms*. PhD thesis, University of Alberta, 2012.

Assaf Hallak and Shie Mannor. Consistent on-line off-policy evaluation. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1372–1383, 2017.

Bo Liu, Ji Liu, Mohammad Ghavamzadeh, Sridhar Mahadevan, and Marek Petrik. Finite-sample analysis of proximal gradient td algorithms. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, pages 504–513, 2015.

Bo Liu, Ji Liu, Mohammad Ghavamzadeh, Sridhar Mahadevan, and Marek Petrik. Proximal gradient temporal difference learning algorithms. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 4195–4199, 2016.

Bo Liu, Ian Gemp, Mohammad Ghavamzadeh, Ji Liu, Sridhar Mahadevan, and Marek Petrik. Proximal gradient temporal difference learning: Stable reinforcement learning with polynomial sample complexity. *Journal of Artificial Intelligence Research*, 63:461–494, 2018.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1054–1062, 2016.

Yangchen Pan, Adam White, and Martha White. Accelerated gradient temporal difference learning. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence*, pages 2464–2470, 2017.

Doina Precup, Richard S Sutton, and Sanjoy Dasgupta. Off-policy temporal-difference learning with function approximation. In *Proceedings of the 18th International Conference on Machine Learning*, pages 417–424, 2001.

Bruno Scherrer. Should one compute the temporal difference fix point or minimize the bellman residual? the unified oblique projection view. In *Proceedings of the 27th International Conference on Machine Learning*, pages 959–966, 2010.

Bruno Scherrer, Mohammad Ghavamzadeh, Victor Gabillon, Boris Lesner, and Matthieu Geist. Approximate modified policy iteration and its application to the game of tetris. *Journal of Machine Learning Research*, 16:1629–1676, 2015.

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

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

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

Richard S Sutton, A Rupam Mahmood, and Martha White. An emphatic approach to the problem of off-policy temporal-difference learning. *The Journal of Machine Learning Research*, 17(1):2603–2631, 2016.

R.S. Sutton, H.R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th International Conference on Machine Learning*, pages 993–1000, 2009.

Christophe Thiery and Bruno Scherrer. Improvements on learning tetris with cross entropy. *International Computer Games Association Journal*, 32(1):23–33, 2009.

Ahmed Touati, Pierre-Luc Bacon, Doina Precup, and Pascal Vincent. Convergent tree backup and retrace with function approximation. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4955–4964, 2018.

John N Tsitsiklis and Benjamin Van Roy. Analysis of temporal-diffference learning with function approximation. In *Advances in neural information processing systems*, pages 1075–1081, 1997.

Shangtong Zhang and Shimon Whiteson. Truncated emphatic temporal difference methods for prediction and control. *arXiv:2108.05338*, 2021.

Shangtong Zhang, Bo Liu, Hengshuai Yao, and Shimon Whiteson. Provably convergent two-timescale off-policy actor-critic with function approximation. In *Proceedings of the 37th International Conference on Machine Learning*, pages 11204–11213, 2020.

Shangtong Zhang, Yi Wan, Richard S Sutton, and Shimon Whiteson. Average-reward off-policy policy evaluation with function approximation. In *Proceedings of the 38th International Conference on Machine Learning*, pages 12578–12588, 2021.