Is Minimizing Errors the Only Option for Value-based Reinforcement Learning?

Anonymous Authors[1]

Abstract

In the regression task of supervised learning, we need to minimize the error and trade off the variance. Drawing on this idea, the existing research on value-based reinforcement learning also minimizes the error. However, is error minimization really the only option for value-based reinforcement learning? We can easily observe that the policy on action choosing probabilities is often related to the relative values, and has nothing to do with their absolute values. Based on this observation, we propose the objective of variance minimization instead of error minimization, derive on-policy and off-policy algorithms respectively, and conduct an analysis of the convergence rate and experiments. The experimental results show that our proposed variance minimization algorithms converge much faster.

1. Introduction

Reinforcement learning can be mainly divided into two categories: value-based reinforcement learning and policy gradient-based reinforcement learning. This paper focuses on temporal difference learning based on linear approximated valued functions. Its research is usually divided into two steps: the first step is to establish the convergence of the algorithm, and the second step is to accelerate the algorithm.

In terms of stability, Sutton (1988) established the convergence of on-policy TD(0), and Tsitsiklis & Van Roy (1997) established the convergence of on-policy TD(λ). However, “The deadly triad” consisting of off-policy learning, bootstrapping, and function approximation makes the stability a difficult problem (Sutton & Barto, 2018). To solve this problem, convergent off-policy temporal difference learning algorithms are proposed, e.g., BR (Baird et al., 1995), GTD (Sutton et al., 2008), GTD2 and TDC (Sutton et al., 2009), ETD (Sutton et al., 2016), and MRetrace (Chen et al., 2023).

In terms of acceleration, Hackman (2012) proposed Hybrid TD algorithm with on-policy matrix. Liu et al. (2015; 2016; 2018) proposed true stochastic algorithms, i.e., GTD-MP and GTD2-MP, from a convex-concave saddle-point formulation. Second-order methods are used to accelerate TD learning, e.g., Quasi Newton TD (Givchi & Palhang, 2015) and accelerated TD (ATD) (Pan et al., 2017). Hallak et al. (2016) introduced an new parameter to reduce variance for ETD. Zhang & Whiteson (2022) proposed truncated ETD with a lower variance. Variance Reduced TD with direct variance reduction technique (Johnson & Zhang, 2013) is proposed by (Korda & La, 2015) and analysed by (Xu et al., 2019). How to further improve the convergence rates of reinforcement learning algorithms is currently still an open problem.

Algorithm stability is prominently reflected in the changes to the objective function, transitioning from mean squared errors (MSE) (Sutton & Barto, 2018) to mean squared bellman errors (MSBE) (Baird et al., 1995), then to norm of the expected TD update (Sutton et al., 2009), and further to mean squared projected Bellman errors (MSPBE) (Sutton et al., 2009). On the other hand, algorithm acceleration is more centered around optimizing the iterative update formula of the algorithm itself without altering the objective function, thereby speeding up the convergence rate of the algorithm. The emergence of new optimization objective functions often leads to the development of novel algorithms. The introduction of new algorithms, in turn, tends to inspire researchers to explore methods for accelerating algorithms, leading to the iterative creation of increasingly superior algorithms.

The kernel loss function can be optimized using standard gradient-based methods, addressing the issue of double sampling in residual gradient algorithm (Feng et al., 2019). It ensures convergence in both on-policy and off-policy scenarios. The logistic bellman error is convex and smooth in the action-value function parameters, with bounded gradients (Bas-Serrano et al., 2021). In contrast, the squared Bellman error is not convex in the action-value function parameters, and RL algorithms based on recursive optimization using it are known to be unstable.

It is necessary to propose a new objective function, but the mentioned objective functions above are all some form of

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

error. Is minimizing error the only option for value-based reinforcement learning?

Error can be decomposed into bias, variance and unavoidable noise. Among them, bias measures the difference between the predicted values of the model and the true values, reflecting the model's fitting ability. Variance, on the other hand, quantifies the model's sensitivity to different training data, indicating its stability and generalization ability. Balancing bias and variance is important, as they represent trade-offs (Zhou, 2021). In the context of this paper, where only a linear model is considered and the model complexity is not adjusted, it is difficult to improve the bias. High bias indicates that the model poorly fits the training data, resulting in underfitting. In supervised learning, high bias is generally considered unacceptable.

However, in reinforcement learning, high bias may be acceptable in certain cases. This is due to the observation that policies based on value functions, such as greedy, ϵ-greedy, and softmax policies, often rely on the relative values of action values rather than their absolute values when selecting different actions.

Based on this observation, we propose alternate objective functions instead of minimizing errors. We minimize Variance of Bellman Error (VBE) and Variance of Projected Bellman Error (VPBE), and derive Variance Minimization (VM) algorithms. These algorithms preserve the invariance of the optimal policy, but significantly reduce the variance of gradient estimation, and thus hastening convergence.

The contributions of this paper are as follows: (1) Introduction of novel objective functions based on the invariance of the optimal policy. (2) Derived two algorithms, one on-policy and one off-policy. (3) Proof of their convergence. (4) Analysis of the convergence rate of on-policy algorithm. (5) Experiments demonstrating the faster convergence speed of the proposed algorithms.

2. Preliminaries

Reinforcement learning agent interacts with environment, observes state, takes sequential decision makings to influence environment, and obtains rewards. Consider an infinite-horizon discounted Markov Decision Process (MDP), defined by a tuple $\langle S, A, R, P, \gamma\rangle$, where $S = \{1, 2, \ldots, N\}$ is a finite set of states of the environment; A is a finite set of actions of the agent; $R : S \times A \times S \to \mathbb{R}$ is a bounded deterministic reward function; $P : S \times A \times S \to [0, 1]$ is the transition probability distribution; and $\gamma \in (0, 1)$ is the discount factor (Sutton & Barto, 2018). Due to the requirements of online learning, value iteration based on sampling is considered in this paper. In each sampling, an experience (or transition) $\langle s, a, s', r\rangle$ is obtained.

A policy is a mapping $\pi : S \times A \to [0, 1]$. The goal of the agent is to find an optimal policy π^* to maximize the expectation of a discounted cumulative rewards in a long period. State value function $V^\pi(s)$ for a stationary policy π is defined as:

$$V^\pi(s) = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_k | s_0 = s].$$

Linear value function for state $s \in S$ is defined as:

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

where $\theta := (\theta_1, \theta_2, \ldots, \theta_m)^\top \in \mathbb{R}^m$ is a parameter vector, $\phi := (\phi_1, \phi_2, \ldots, \phi_m)^\top \in \mathbb{R}^m$ is a feature function defined on state space S, and m is the feature size.

Tabular temporal difference (TD) learning (Sutton & Barto, 2018) has been successfully applied to small-scale problems. To deal with the well-known curse of dimensionality of large scale MDPs, value function is usually approximated by a linear model, kernel methods, decision trees, or neural networks, etc. This paper focuses on the linear model, where features are usually hand coded by domain experts.

TD learning can also be used to find optimal strategies. The problem of finding an optimal policy is often called the control problem. Two popular TD methods are Sarsa and Q-leaning. The former is an on-policy TD control, while the latter is an off-policy control.

It is well known that TDC algorithm (Sutton et al., 2009) guarantees convergence under off-policy conditions while the off-policy TD algorithm may diverge. The objective function of TDC is MSPBE. TDC is essentially an adjustment or correction of the TD update so that it follows the gradient of the MSPBE objective function. In the context of the TDC algorithm, the control algorithm is known as Greedy-GQ(λ) (Sutton et al., 2009). When λ is set to 0, it is denoted as GQ(0).

3. Variance Minimization Algorithms

3.1. Motivation

In reinforcement learning, bias is acceptable, while in supervised learning it is not. As shown in Table 1, although there is a bias between the true value and the predicted value, action a_3 is still chosen under the greedy-policy. On the contrary, supervised learning is usually used to predict temperature, humidity, morbidity, etc. If the bias is too large, the consequences could be serious.

In addition, reward shaping can significantly speed up the learning by adding a shaping reward $F(s, s')$ to the original reward r, where $F(s, s')$ is the general form of any state-based shaping reward. Static potential-based reward shaping

Table 1. Classification accuracies for naive Bayes and flexible Bayes on various data sets.

ACTION	Q VALUE	Q VALUE WITH BIAS
$Q(s,a_0)$	1	5
$Q(s,a_1)$	2	6
$Q(s,a_2)$	3	7
$Q(s,a_3)$	4	8
$\arg\min_a Q(s,a)$	a_3	a_3

(Static PBRS) maintains the policy invariance if the shaping reward follows from $F(s,s') = \gamma f(s') - f(s)$ (Ng et al., 1999).

This means that we can make changes to the TD error $\delta = r + \gamma\theta^\top\phi' - \theta^\top\phi$ while still ensuring the invariance of the optimal policy,

$$\delta - \omega = r + \gamma\theta^\top\phi' - \theta^\top\phi - \omega,$$

where ω is a constant, acting as a static PBRS. This also means that algorithms with the optimization goal of minimizing errors, after introducing reward shaping, may result in larger or smaller bias. Fortunately, as discussed above, bias is acceptable in reinforcement learning. However, the problem is that selecting an appropriate ω requires expert knowledge. This forces us to learn ω dynamically, i.e., $\omega = \omega_t$ and dynamic PBRS can also maintain the policy invariance if the shaping reward is $F(s,t,s',t') = \gamma f(s',t') - f(s,t)$, where t is the time-step the agent reaches in state s (Devlin & Kudenko, 2012). However, this result requires the convergence guarantee of the dynamic potential function $f(s,t)$. If $f(s,t)$ does not converge as the time-step $t \to \infty$, the Q-values of dynamic PBRS are not guaranteed to converge.

Let $f_{\omega_t}(s) = \frac{\omega_t}{\gamma - 1}$. Thus, $F_{\omega_t}(s,s') = \gamma f_{\omega_t}(s') - f_{\omega_t}(s) = \omega_t$ is a dynamic PBRS. And if ω converges finally, the dynamic potential function $f(s,t)$ will converge. Bias is the expected difference between the predicted value and the true value. Therefore, under the premise of bootstrapping, we first think of letting $\omega \doteq \mathbb{E}[\mathbb{E}[\delta|s]] = \mathbb{E}[\delta]$.

As we all know, the optimization process of linear TD(0) (semi-gradient) and linear TDC are as follows, respectively:

$$\theta^* = \arg\min_\theta \mathbb{E}[(\mathbb{E}[\delta|s])^2],$$

and

$$\theta^* = \arg\min_\theta \mathbb{E}[\delta\phi]^\top \mathbb{E}[\phi\phi^\top]^{-1}\mathbb{E}[\delta\phi].$$

As a result, two novel objective functions and their corresponding algorithms are proposed, where ω is subsequently proven to converge, meaning that these two algorithms can maintain the invariance of the optimal strategy.

3.2. Variance Minimization TD Learning: VMTD

For on-policy learning, a novel objective function, Variance of Bellman Error (VBE), is proposed as follows:

$$\begin{array}{ccl} \arg\min_\theta \text{VBE}(\theta) & = & \arg\min_\theta \mathbb{E}[(\mathbb{E}[\delta|s] - \mathbb{E}[\mathbb{E}[\delta|s]])^2] \\ & = & \arg\min_{\theta,\omega} \mathbb{E}[(\mathbb{E}[\delta|s] - \omega)^2]. \end{array} \tag{2}$$

Clearly, it is no longer to minimize Bellman errors.

First, the parameter ω is derived directly based on stochastic gradient descent:

$$\omega_{k+1} \leftarrow \omega_k + \beta_k(\delta_k - \omega_k), \tag{3}$$

where δ_k is the TD error as follows:

$$\delta_k = r + \gamma\theta_k^\top\phi'_k - \theta_k^\top\phi_k. \tag{4}$$

Then, based on stochastic semi-gradient descent, the update of the parameter θ is as follows:

$$\theta_{k+1} \leftarrow \theta_k + \alpha_k(\delta_k - \omega_k)\phi_k. \tag{5}$$

The pseudocode of the VMTD algorithm is shown in Algorithm 1.

For control tasks, two extensions of VMTD are named VMSarsa and VMQ respectively, and the update formulas are shown below:

$$\theta_{k+1} \leftarrow \theta_k + \alpha_k(\delta_k - \omega_k)\phi(s_k,a_k). \tag{6}$$

and

$$\omega_{k+1} \leftarrow \omega_k + \beta_k(\delta_k - \omega_k), \tag{7}$$

where δ_k delta in VMSarsa is:

$$\delta_k = r_{k+1} + \gamma\theta_k^\top\phi(s_{k+1},a_{k+1}) - \theta_k^\top\phi(s_k,a_k), \tag{8}$$

and δ_k delta in VMQ is:

$$\delta_k = r_{k+1} + \gamma\max_{a\in A}\theta_k^\top\phi(s_{k+1},a) - \theta_k^\top\phi(s_k,a_k). \tag{9}$$

3.3. Variance Minimization TDC Learning: VMTDC

For off-policy learning, we employ a projection operator. The objective function is called Variance of Projected Bellman error (VPBE), and the corresponding algorithm is called VMTDC.

$$\begin{array}{ccl} \text{VPBE}(\theta) & = & \mathbb{E}[(\delta - \mathbb{E}[\delta])\phi]^\top\mathbb{E}[\phi\phi^\top]^{-1}\mathbb{E}[(\delta - \mathbb{E}[\delta])\phi] \\ & = & \mathbb{E}[(\delta - \omega)\phi]^\top\mathbb{E}[\phi\phi^\top]^{-1}\mathbb{E}[(\delta - \omega)\phi], \end{array} \tag{10}$$

where ω is used to estimate $\mathbb{E}[\delta]$, i.e., $\omega \doteq \mathbb{E}[\delta]$.

The derivation process of the VMTDC algorithm is the same as that of the TDC algorithm, the only difference is that the

Algorithm 1 VMTD algorithm with linear function approximation in the on-policy setting

```
Input: $\theta_0$, $\omega_0$, $\gamma$, learning rate $\alpha_t$ and $\beta_t$
repeat
  For any episode, initialize $\theta_0$ arbitrarily, $\omega_0$ to 0, $\gamma \in (0,1]$, and $\alpha_t$ and $\beta_t$ are constant.
  for $t = 0$ to $T-1$ do
    Take $A_t$ from $S_t$ according to policy $\mu$, and arrive at $S_{t+1}$
    Observe sample ($S_t$,$R_{t+1}$,$S_{t+1}$) at time step $t$ (with their corresponding state feature vectors)
    $\delta_t = R_{t+1} + \gamma\theta_t^\top\phi'_t - \theta_t^\top\phi_t$
    $\theta_{t+1} \leftarrow \theta_t + \alpha_t(\delta_t - \omega_t)\phi_t$
    $\omega_{t+1} \leftarrow \omega_t + \beta_t(\delta_t - \omega_t)$
    $S_t = S_{t+1}$
  end for
until terminal episode
```

original δ is replaced by $\delta - \omega$. Therefore, we can easily get the updated formula of VMTDC, as follows:

$$\theta_{k+1} \leftarrow \theta_k + \alpha_k[(\delta_k - \omega_k)\phi(s_k) - \gamma\phi(s_{k+1})(\phi^\top(s_k)u_k)], \tag{11}$$

$$u_{k+1} \leftarrow u_k + \zeta_k[\delta_k - \omega_k - \phi^\top(s_k)u_k]\phi(s_k), \tag{12}$$

and

$$\omega_{k+1} \leftarrow \omega_k + \beta_k(\delta_k - \omega_k), \tag{13}$$

The pseudocode of the VMTDC algorithm for importance-sampling scenario is shown in Algorithm 2 of Appendix A.2.

Now, we will introduce the improved version of the GQ(0) algorithm, named VMGQ(0):

$$\begin{aligned}\theta_{k+1} \leftarrow \theta_k \quad &+ \quad \alpha_k[(\delta_k - \omega_k)\phi(s_k, a_k) \\ &- \quad \gamma\phi(s_{k+1}, A^*_{k+1})(\phi^\top(s_k, a_k)u_k)],\end{aligned} \tag{14}$$

$$u_{k+1} \leftarrow u_k + \zeta_k[(\delta_k - u_k) - \phi^\top(s_k, a_k)u_k]\phi(s_k, a_k), \tag{15}$$

and

$$\omega_{k+1} \leftarrow \omega_k + \beta_k(\delta_k - \omega_k), \tag{16}$$

where δ_k is (9) and $A^*_{k+1} = \arg\max_a(\theta_k^\top\phi(s_{k+1}, a))$.

4. Theoretical Analysis

The purpose of this section is to establish the stabilities of the VMTD algorithm and the VMTDC algorithm, and also presents a corollary on the convergence rate of VMTD.

Theorem 4.1. *(Convergence of VMTD). In the case of on-policy learning, consider the iterations (3) and (5) with (4) of VMTD. Let the step-size sequences α_k and β_k, $k \geq 0$ satisfy in this case $\alpha_k, \beta_k > 0$, for all k, $\sum_{k=0}^{\infty}\alpha_k = \sum_{k=0}^{\infty}\beta_k = \infty$, $\sum_{k=0}^{\infty}\alpha_k^2 < \infty$, $\sum_{k=0}^{\infty}\beta_k^2 < \infty$, and $\alpha_k = o(\beta_k)$. Assume that (ϕ_k, r_k, ϕ'_k) is an i.i.d. sequence with uniformly bounded second moments, where ϕ_k and ϕ'_k are sampled from the same Markov chain. Let $A = \mathrm{Cov}(\phi, \phi - \gamma\phi')$, $b = \mathrm{Cov}(r, \phi)$. Assume that matrix A is non-singular. Then the parameter vector θ_k converges with probability one to $A^{-1}b$.*

Proof. The proof is based on Borkar's Theorem for general stochastic approximation recursions with two time scales (Borkar, 1997).

A new one-step linear TD solution is defined as:

$$0 = \mathbb{E}[(\delta - \mathbb{E}[\delta])\phi] = -A\theta + b.$$

Thus, the VMTD's solution is $\theta_{\text{VMTD}} = A^{-1}b$.

First, note that recursion (5) can be rewritten as

$$\theta_{k+1} \leftarrow \theta_k + \beta_k\xi(k),$$

where

$$\xi(k) = \frac{\alpha_k}{\beta_k}(\delta_k - \omega_k)\phi_k$$

Due to the settings of step-size schedule $\alpha_k = o(\beta_k)$, $\xi(k) \rightarrow 0$ almost surely as $k \rightarrow \infty$. That is the increments in iteration (3) are uniformly larger than those in (5), thus (3) is the faster recursion. Along the faster time scale, iterations of (3) and (5) are associated to ODEs system as follows:

$$\dot{\theta}(t) = 0, \tag{17}$$

$$\dot{\omega}(t) = \mathbb{E}[\delta_t|\theta(t)] - \omega(t). \tag{18}$$

Based on the ODE (17), $\theta(t) \equiv \theta$ when viewed from the faster timescale. By the Hirsch lemma (Hirsch, 1989), it follows that $||\theta_k - \theta|| \rightarrow 0$ a.s. as $k \rightarrow \infty$ for some θ that depends on the initial condition θ_0 of recursion (5). Thus, the ODE pair (17)-(18) can be written as

$$\dot{\omega}(t) = \mathbb{E}[\delta_t|\theta] - \omega(t). \tag{19}$$

Consider the function $h(\omega) = \mathbb{E}[\delta|\theta] - \omega$, i.e., the driving vector field of the ODE (19). It is easy to find that the function h is Lipschitz with coefficient -1. Let $h_\infty(\cdot)$ be the function defined by $h_\infty(\omega) = \lim_{x\rightarrow\infty}\frac{h(x\omega)}{x}$. Then $h_\infty(\omega) = -\omega$, is well-defined. For (19), $\omega^* = \mathbb{E}[\delta|\theta]$ is the unique globally asymptotically stable equilibrium. For the ODE

$$\dot{\omega}(t) = h_\infty(\omega(t)) = -\omega(t), \tag{20}$$

apply $\vec{V}(\omega) = (-\omega)^\top(-\omega)/2$ as its associated strict Liapunov function. Then, the origin of (20) is a globally asymptotically stable equilibrium.

Consider now the recursion (3). Let $M_{k+1} = (\delta_k - \omega_k) - \mathbb{E}[(\delta_k - \omega_k)|\mathcal{F}(k)]$, where $\mathcal{F}(k) = \sigma(\omega_l, \theta_l, l \leq k; \phi_s, \phi'_s, r_s, s < k)$, $k \geq 1$ are the sigma fields generated by $\omega_0, \theta_0, \omega_{l+1}, \theta_{l+1}, \phi_l, \phi'_l$, $0 \leq l < k$. It is easy to verify that $M_{k+1}, k \geq 0$ are integrable random variables that satisfy $\mathbb{E}[M_{k+1}|\mathcal{F}(k)] = 0, \forall k \geq 0$. Because ϕ_k, r_k, and ϕ'_k have uniformly bounded second moments, it can be seen that for some constant $c_1 > 0, \forall k \geq 0$,

$$\mathbb{E}[||M_{k+1}||^2|\mathcal{F}(k)] \leq c_1(1 + ||\omega_k||^2 + ||\theta_k||^2).$$

Now Assumptions (A1) and (A2) of (Borkar & Meyn, 2000) are verified. Furthermore, Assumptions (TS) of (Borkar & Meyn, 2000) is satisfied by our conditions on the step-size sequences α_k, β_k. Thus, by Theorem 2.2 of (Borkar & Meyn, 2000) we obtain that $||\omega_k - \omega^*|| \to 0$ almost surely as $k \to \infty$.

Consider now the slower time scale recursion (5). Based on the above analysis, (5) can be rewritten as

$$\theta_{k+1} \leftarrow \theta_k + \alpha_k(\delta_k - \mathbb{E}[\delta_k|\theta_k])\phi_k.$$

Let $\mathcal{G}(k) = \sigma(\theta_l, l \leq k; \phi_s, \phi'_s, r_s, s < k)$, $k \geq 1$ be the sigma fields generated by $\theta_0, \theta_{l+1}, \phi_l, \phi'_l$, $0 \leq l < k$. Let $Z_{k+1} = Y_t - \mathbb{E}[Y_t|\mathcal{G}(k)]$, where

$$Y_k = (\delta_k - \mathbb{E}[\delta_k|\theta_k])\phi_k.$$

Consequently,

$$\begin{array}{rcl}\mathbb{E}[Y_t|\mathcal{G}(k)] & = & \mathbb{E}[(\delta_k - \mathbb{E}[\delta_k|\theta_k])\phi_k|\mathcal{G}(k)] \\ & = & \mathbb{E}[\delta_k\phi_k|\theta_k] - \mathbb{E}[\mathbb{E}[\delta_k|\theta_k]\phi_k] \\ & = & \mathbb{E}[\delta_k\phi_k|\theta_k] - \mathbb{E}[\delta_k|\theta_k]\mathbb{E}[\phi_k] \\ & = & \mathrm{Cov}(\delta_k|\theta_k, \phi_k),\end{array}$$

where $\mathrm{Cov}(\cdot, \cdot)$ is a covariance operator.

Thus,

$$Z_{k+1} \quad = \quad (\delta_k - \mathbb{E}[\delta_k|\theta_k])\phi_k - \mathrm{Cov}(\delta_k|\theta_k, \phi_k).$$

It is easy to verify that $Z_{k+1}, k \geq 0$ are integrable random variables that satisfy $\mathbb{E}[Z_{k+1}|\mathcal{G}(k)] = 0$, $\forall k \geq 0$. Also, because ϕ_k, r_k, and ϕ'_k have uniformly bounded second moments, it can be seen that for some constant $c_2 > 0$, $\forall k \geq 0$,

$$\mathbb{E}[||Z_{k+1}||^2|\mathcal{G}(k)] \leq c_2(1 + ||\theta_k||^2).$$

Consider now the following ODE associated with (5):

$$\begin{array}{rcl}\dot{\theta}(t) & = & \mathrm{Cov}(\delta|\theta(t), \phi) \\ & = & \mathrm{Cov}(r + (\gamma\phi' - \phi)^\top\theta(t), \phi) \\ & = & \mathrm{Cov}(r, \phi) - \mathrm{Cov}(\theta(t)^\top(\phi - \gamma\phi'), \phi) \\ & = & \mathrm{Cov}(r, \phi) - \theta(t)^\top\mathrm{Cov}(\phi - \gamma\phi', \phi) \\ & = & \mathrm{Cov}(r, \phi) - \mathrm{Cov}(\phi - \gamma\phi', \phi)^\top\theta(t) \\ & = & \mathrm{Cov}(r, \phi) - \mathrm{Cov}(\phi, \phi - \gamma\phi')\theta(t) \\ & = & -A\theta(t) + b.\end{array} \tag{21}$$

Let $\vec{h}(\theta(t))$ be the driving vector field of the ODE (21).

$$\vec{h}(\theta(t)) = -A\theta(t) + b.$$

Consider the cross-covariance matrix,

$$\begin{array}{rcl}A & = & \mathrm{Cov}(\phi, \phi - \gamma\phi') \\ & = & \frac{\mathrm{Cov}(\phi,\phi)+\mathrm{Cov}(\phi-\gamma\phi',\phi-\gamma\phi')-\mathrm{Cov}(\gamma\phi',\gamma\phi')}{2} \\ & = & \frac{\mathrm{Cov}(\phi,\phi)+\mathrm{Cov}(\phi-\gamma\phi',\phi-\gamma\phi')-\gamma^2\mathrm{Cov}(\phi',\phi')}{2} \\ & = & \frac{(1-\gamma^2)\mathrm{Cov}(\phi,\phi)+\mathrm{Cov}(\phi-\gamma\phi',\phi-\gamma\phi')}{2},\end{array} \tag{22}$$

where we eventually used $\mathrm{Cov}(\phi', \phi') = \mathrm{Cov}(\phi, \phi)$ [1]. Note that the covariance matrix $\mathrm{Cov}(\phi, \phi)$ and $\mathrm{Cov}(\phi - \gamma\phi', \phi - \gamma\phi')$ are semi-positive definite. Then, the matrix A is semi-positive definite because A is linearly combined by two positive-weighted semi-positive definite matrice (22). Furthermore, A is nonsingular due to the assumption. Hence, the cross-covariance matrix A is positive definite.

Therefore, $\theta^* = A^{-1}b$ can be seen to be the unique globally asymptotically stable equilibrium for ODE (21). Let $\vec{h}_\infty(\theta) = \lim_{r\to\infty} \frac{\vec{h}(r\theta)}{r}$. Then $\vec{h}_\infty(\theta) = -A\theta$ is well-defined. Consider now the ODE

$$\dot{\theta}(t) = -A\theta(t). \tag{23}$$

The ODE (23) has the origin as its unique globally asymptotically stable equilibrium. Thus, the assumption (A1) and (A2) are verified. □

Theorem 3 in (Dalal et al., 2020) provides a general conclusion on the convergence speed of all linear two-timescale algorithms. VMTD satisfies the assumptions of this theorem, leading to the following corollary.

Corollary 4.2. *Consider the Sparsely Projected variant of VMTD. Then, for $\alpha_k = 1/(k+1)^\alpha$, $\beta_k = 1/(k+1)^\beta$, $0 < \beta < \alpha < 1$, $p > 1$, with probility larger than $1 - \tau$, for all $k \geq N_3$, we have*

$$||\theta'_k - \theta^*|| \leq C_{3,\theta}\frac{\sqrt{\ln(4d_1^2(k+1)^p/\tau)}}{(k+1)^{\alpha/2}} \tag{24}$$

$$||\omega'_n - \omega^*|| \leq C_{3,\omega}\frac{\sqrt{\ln(4d_2^2(k+1)^p/\tau)}}{(k+1)^{\omega/2}}, \tag{25}$$

where d_1 and d_2 represent the dimensions of θ and ω, respectively. For VMTD, $d_2 = 1$. The meanings of N_3,$C_{3,\theta}$ and $C_{3,\omega}$ are explained in (Dalal et al., 2020). The formulas for θ'_k and ω'_n can be found in (30) and (31).

Theorem 4.3. *(Convergence of VMTDC). In the case of off-policy learning, consider the iterations (13), (12) and (11) of VMTDC. Let the step-size sequences α_k, ζ_k and β_k, $k \geq 0$ satisfy in this case $\alpha_k, \zeta_k, \beta_k > 0$, for all k, $\sum_{k=0}^{\infty} \alpha_k = \sum_{k=0}^{\infty} \beta_k = \sum_{k=0}^{\infty} \zeta_k = \infty$, $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$, $\sum_{k=0}^{\infty} \zeta_k^2 < \infty$, $\sum_{k=0}^{\infty} \beta_k^2 < \infty$, and $\alpha_k = o(\zeta_k)$, $\zeta_k =$*

[1] The covariance matrix $\mathrm{Cov}(\phi', \phi')$ is equal to the covariance matrix $\mathrm{Cov}(\phi, \phi)$ if the initial state is re-reachable or initialized randomly in a Markov chain for on-policy update.

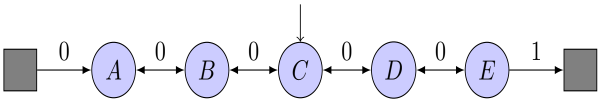


Figure 1. Random walk.

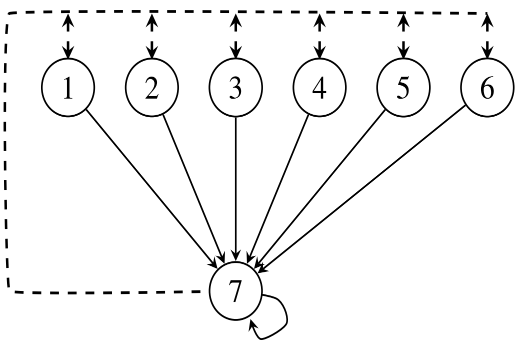


Figure 2. 7-state version of Baird's off-policy counterexample.

$o(\beta_k)$. *Assume that* (ϕ_k, r_k, ϕ'_k) *is an i.i.d. sequence with uniformly bounded second moments. Let* $A = \mathrm{Cov}(\phi, \phi - \gamma\phi')$*,* $b = \mathrm{Cov}(r, \phi)$*, and* $C = \mathbb{E}[\phi\phi^\top]$*. Assume that* A *and* C *are non-singular matrices. Then the parameter vector* θ_k *converges with probability one to* $A^{-1}b$*.*

Please refer to the appendix A.2 for detailed proof process.

5. Experimental Studies

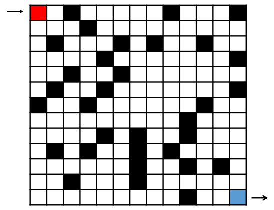
This section assesses algorithm performance through experiments, which are divided into policy evaluation experiments and control experiments.

5.1. Testing Tasks

Random-walk: as shown in Figure 1, all episodes start in the center state, C, and proceed to left or right by one state on each step, equiprobably. Episodes terminate either on the extreme left or the extreme right, and get a reward of $+1$ if terminate on the right, or 0 in the other case. In this task, the true value for each state is the probability of starting from that state and terminating on the right (Sutton & Barto, 2018). Thus, the true values of states from A to E are $\frac{1}{6}, \frac{2}{6}, \frac{3}{6}, \frac{4}{6}, \frac{5}{6}$, respectively. The discount factor $\gamma = 1.0$. There are three standard kinds of features for random-walk problems: tabular feature, inverted feature and dependent feature (Sutton et al., 2009). The feature matrices corresponding to three random walks are shown in Appendix B. Conduct experiments using an on-policy approach in the Random-walk environment.

Baird's off-policy counterexample: This task is well known as a counterexample, in which TD diverges (Baird et al., 1995; Sutton et al., 2009). As shown in Figure 2, reward for each transition is zero. Thus the true values are zeros for all states and for any given policy. The behaviour policy chooses actions represented by solid lines with a probability of $\frac{1}{7}$ and actions represented by dotted lines with a probability of $\frac{6}{7}$. The target policy is expected to choose the solid line with more probability than $\frac{1}{7}$, and it chooses the solid line with probability of 1 in this paper. The discount factor $\gamma = 0.99$, and the feature matrix is defined in Appendix B (Baird et al., 1995; Sutton et al., 2009; Maei, 2011).

Maze: The learning agent should find a shortest path from the upper left corner to the lower right corner. In each state, there are four alternative actions: up, $down$, $left$, and $right$, which takes the agent deterministically to the corresponding neighbour state, except when a movement is blocked by an obstacle or the edge of the maze. Rewards are -1 in all transitions until the agent reaches the goal state. The discount factor $\gamma = 0.99$, and states s are represented by tabular features.The maximum number of moves in the game is set to 1000.



The other three control environments: Cliff Walking, Mountain Car, and Acrobot are selected from the gym official website and correspond to the following versions: "CliffWalking-v0", "MountainCar-v0" and "Acrobot-v1". For specific details, please refer to the gym official website. The maximum number of steps for the Mountain Car environment is set to 1000, while the default settings are used for the other two environments. In Mountain car and Acrobot, features are generated by tile coding.

Please, refer to the Appendix B for the selection of learning rates for all experiments.

5.2. Experimental Results and Analysis

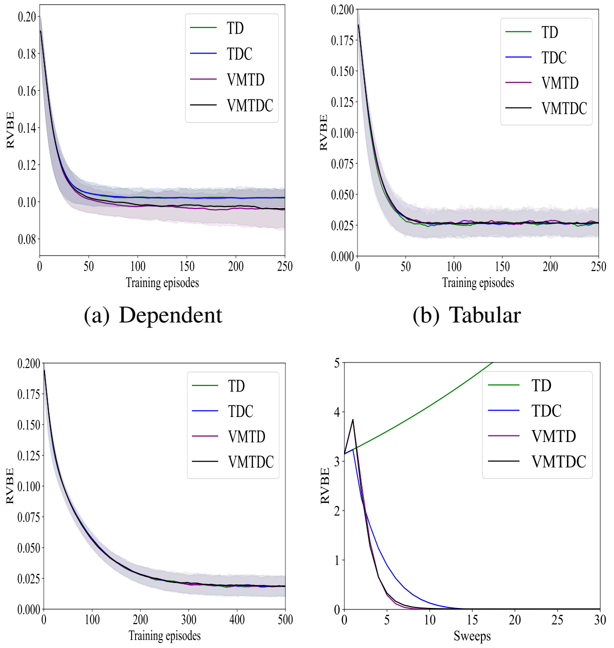
For policy evaluation experiments, compare the performance of the VMTD, VMTDC, TD, and TDC algorithms. The vertical axis is unified as RVBE.

For policy evaluation experiments, the criteria for evaluating algorithms vary. The objective function minimized by our proposed new algorithm differs from that of other algorithms. Therefore, to ensure fairness in comparisons, this study only contrasts algorithm experiments in controlled settings.

This study will compare the performance of Sarsa, Q-learning, GQ(0), AC, VMSarsa, VMQ, and VMGQ(0) in four control environments.

Table 2. Difference between R-learning and tabular VMQ.

algorithms	update formula
R-learning	$Q_{k+1}(s,a) \leftarrow Q_k(s,a) + \alpha_k(r_{k+1} - m_k + \max_{b\in A} Q_k(s,b) - Q_k(s,a))$ $m_{k+1} \leftarrow m_k + \beta_k(r_{k+1} + \max_{b\in A} Q_k(s,b) - Q_k(s,a) - m_k)$
tabular VMQ	$Q_{k+1}(s,a) \leftarrow Q_k(s,a) + \alpha_k(r_{k+1} + \gamma \max_{b\in A} Q_k(s,b) - Q_k(s,a) - \omega_k)$ $\omega_{k+1} \leftarrow \omega_k + \beta_k(r_{k+1} + \gamma \max_{b\in A} Q_k(s,b) - Q_k(s,a) - \omega_k)$



(a) Dependent

(b) Tabular

(c) Inverted

(d) counterexample

Figure 3. Learning curses of four evaluation environments.

The learning curves of the algorithms corresponding to policy evaluation experiments and control experiments are shown in Figures 3 and 4, respectively. The shaded area in Figure 3, 4 represents the standard deviation (std).

In the random-walk tasks, VMTD and VMTDC exhibit excellent performance, outperforming TD and TDC in the case of dependent random-walk.

In the 7-state example counter task, TD diverges, while VMTDC converges and performs better than TDC. From the update formula, it can be observed that the VMTD algorithm, like TDC, is also an adjustment or correction of the TD update. What is more surprising is that VMTD also maintains convergence and demonstrates the best performance.

In Maze, Mountain Car, and Acrobot, the convergence speed of VMSarsa, VMQ, and VMGQ(0) has been significantly improved compared to Sarsa, Q-learning, and GQ(0), respectively. The performance of the AC algorithm is at an intermediate level. The performances of VMSarsa, VMQ, and VMGQ(0) in these three experimental environments have no significant differences.

In Cliff Walking, Sarsa and VMSarsa converge to slightly worse solutions compared to other algorithms. The convergence speed of VMSarsa is significantly better than that of Sarsa. The convergence speed of VMGQ(0) and VMQ is better than other algorithms, and the performance of VMGQ(0) is slightly better than that of VMQ.

In summary, the performance of VMSarsa, VMQ, and VMGQ(0) is better than that of other algorithms. In the Cliff Walking environment, the performance of VMGQ(0) is slightly better than that of VMSarsa and VMQ. In the other three experimental environments, the performances of VMSarsa, VMQ, and VMGQ(0) are close.

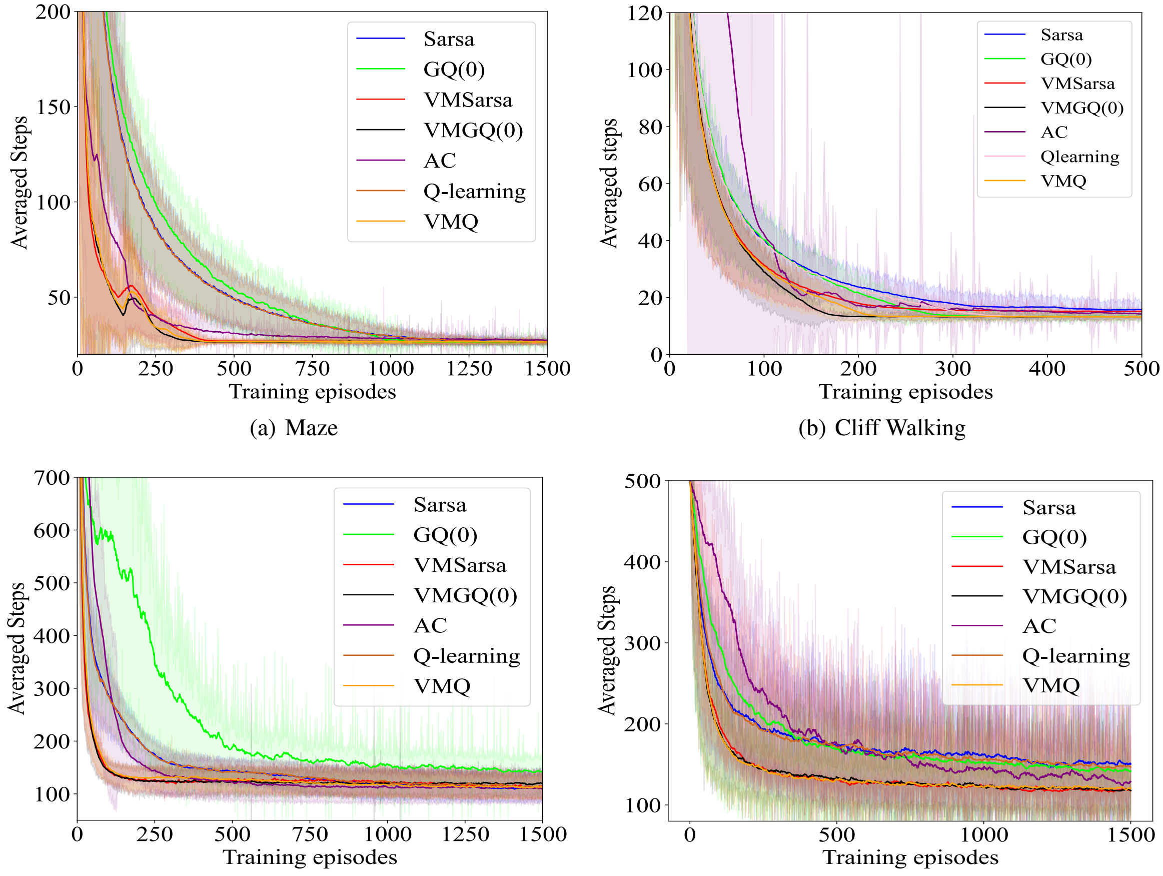
6. Related Work

6.1. Difference between VMQ and R-learning

Tabular VMQ's update formula bears some resemblance to R-learning's update formula. As shown in Table 2, the update formulas of the two algorithms have the following differences:
(1) The goal of the R-learning algorithm (Schwartz, 1993) is to maximize the average reward, rather than the cumulative reward, by learning an estimate of the average reward. This estimate m is then used to update the Q-values. On the contrary, the ω in the tabular VMQ update formula eventually converges to $\mathbb{E}[\delta]$.
(2) When $\gamma = 1$ in the tabular VMQ update formula, the R-learning update formula is formally the same as the tabular VMQ update formula. Therefore, R-learning algorithm can be considered as a special case of VMQ algorithm in form.

6.2. Variance Reduction for TD Learning

The TD with centering algorithm (CTD) (Korda & La, 2015) was proposed, which directly applies variance reduction techniques to the TD algorithm. The CTD algorithm updates its parameters using the average gradient of a batch of Markovian samples and a projection operator. Unfortunately, the authors' analysis of the CTD algorithm contains technical errors. The VRTD algorithm (Xu et al., 2020) is also a variance-reduced algorithm that updates its parame-



(a) Maze

(b) Cliff Walking

(c) Mountain Car

(d) Acrobot

Figure 4. Learning curses of four contral environments.

ters using the average gradient of a batch of i.i.d. samples. The authors of VRTD provide a technically sound analysis to demonstrate the advantages of variance reduction.

6.3. Variance Reduction for Policy Gradient Algorithms

Policy gradient algorithms are a class of reinforcement learning algorithms that directly optimize cumulative rewards. REINFORCE is a Monte Carlo algorithm that estimates gradients through sampling, but may have a high variance. Baselines are introduced to reduce variance and to accelerate learning (Sutton & Barto, 2018). In Actor-Critic, value function as a baseline and bootstrapping are used to reduce variance, also accelerating convergence (Sutton & Barto, 2018). TRPO (Schulman et al., 2015) and PPO (Schulman et al., 2017) use generalized advantage estimation, which combines multi-step bootstrapping and Monte Carlo estimation to reduce variance, making gradient estimation more stable and accelerating convergence.

In Variance Minimization, the incorporation of $\omega \doteq \mathbb{E}[\delta]$ bears a striking resemblance to the use of a baseline in policy gradient methods. The introduction of a baseline in policy gradient techniques does not alter the expected value of the update; rather, it significantly impacts the variance of gradient estimation. The addition of $\omega \doteq \mathbb{E}[\delta]$ in Variance Minimization preserves the invariance of the optimal policy while stabilizing gradient estimation, reducing the variance of gradient estimation, and hastening convergence.

7. Conclusion and Future Work

Value-based reinforcement learning typically aims to minimize error as an optimization objective. As an alternation, this study proposes two new objective functions: VBE and VPBE, and derives an on-policy algorithm: VMTD and an off-policy algorithm: VMTDC. Both algorithms demonstrated superior performance in policy evaluation and control experiments. Future work may include, but are not limited to, (1) analysis of the convergence rate of VMTDC. (2) extensions of VBE and VPBE to multi-step returns. (3) extensions to nonlinear approximations, such as neural networks.

References

Baird, L. et al. Residual algorithms: Reinforcement learning with function approximation. In *Proc. 12th Int. Conf. Mach. Learn.*, pp. 30–37, 1995.

Bas-Serrano, J., Curi, S., Krause, A., and Neu, G. Logistic q-learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 3610–3618, 2021.

Borkar, V. S. Stochastic approximation with two time scales. *Syst. & Control Letters*, 29(5):291–294, 1997.

Borkar, V. S. and Meyn, S. P. The ode method for convergence of stochastic approximation and reinforcement learning. *SIAM J. Control Optim.*, 38(2):447–469, 2000.

Chen, X., Ma, X., Li, Y., Yang, G., Yang, S., and Gao, Y. Modified retrace for off-policy temporal difference learning. In *Uncertainty in Artificial Intelligence*, pp. 303–312. PMLR, 2023.

Dalal, G., Szorenyi, B., and Thoppe, G. A tale of two-timescale reinforcement learning with the tightest finite-time bound. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 3701–3708, 2020.

Devlin, S. and Kudenko, D. Dynamic potential-based reward shaping. In *Proc. 11th Int. Conf. Autonomous Agents and Multiagent Systems*, pp. 433–440, 2012.

Feng, Y., Li, L., and Liu, Q. A kernel loss for solving the bellman equation. In *Advances in Neural Information Processing Systems*, pp. 15430–15441, 2019.

Givchi, A. and Palhang, M. Quasi newton temporal difference learning. In *Asian Conference on Machine Learning*, pp. 159–172, 2015.

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

Hallak, A., Tamar, A., Munos, R., and Mannor, S. Generalized emphatic temporal difference learning: bias-variance analysis. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pp. 1631–1637, 2016.

Hirsch, M. W. Convergent activation dynamics in continuous time networks. *Neural Netw.*, 2(5):331–349, 1989.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pp. 315–323, 2013.

Korda, N. and La, P. On td (0) with function approximation: Concentration bounds and a centered variant with exponential convergence. In *International conference on machine learning*, pp. 626–634. PMLR, 2015.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Liu, B., Liu, J., Ghavamzadeh, M., Mahadevan, S., and Petrik, M. Finite-sample analysis of proximal gradient td algorithms. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, pp. 504–513, 2015.

Liu, B., Liu, J., Ghavamzadeh, M., Mahadevan, S., and Petrik, M. Proximal gradient temporal difference learning algorithms. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 4195–4199, 2016.

Liu, B., Gemp, I., Ghavamzadeh, M., Liu, J., Mahadevan, S., and Petrik, M. Proximal gradient temporal difference learning: Stable reinforcement learning with polynomial sample complexity. *Journal of Artificial Intelligence Research*, 63:461–494, 2018.

Maei, H. R. *Gradient temporal-difference learning algorithms*. PhD thesis, University of Alberta, 2011.

Ng, A. Y., Harada, D., and Russell, S. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proc. 16th Int. Conf. Mach. Learn.*, pp. 278–287, 1999.

Pan, Y., White, A., and White, M. Accelerated gradient temporal difference learning. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence*, pp. 2464–2470, 2017.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International Conference on Machine Learning*, pp. 1889–1897, 2015.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Schwartz, A. A reinforcement learning method for maximizing undiscounted rewards. In *Proc. 10th Int. Conf. Mach. Learn.*, volume 298, pp. 298–305, 1993.

Sutton, R., Maei, H., Precup, D., Bhatnagar, S., Silver, D., Szepesvári, C., and Wiewiora, E. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proc. 26th Int. Conf. Mach. Learn.*, pp. 993–1000, 2009.

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

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

Sutton, R. S., Maei, H. R., and Szepesvári, C. A convergent $o(n)$ temporal-difference algorithm for off-policy learning with linear function approximation. In *Advances in Neural Information Processing Systems*, pp. 1609–1616. Cambridge, MA: MIT Press, 2008.

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

Tsitsiklis, J. N. and Van Roy, B. Analysis of temporal-diffference learning with function approximation. In *Advances in Neural Information Processing Systems*, pp. 1075–1081, 1997.

Xu, T., Wang, Z., Zhou, Y., and Liang, Y. Reanalysis of variance reduced temporal difference learning. In *International Conference on Learning Representations*, 2019.

Xu, T., Wang, Z., Zhou, Y., and Liang, Y. Reanalysis of variance reduced temporal difference learning. *arXiv preprint arXiv:2001.01898*, 2020.

Zhang, S. and Whiteson, S. Truncated emphatic temporal difference methods for prediction and control. *The Journal of Machine Learning Research*, 23(1):6859–6917, 2022.

Zhou, Z.-H. *Machine learning*. Springer Nature, 2021.

A. Relevant proofs

A.1. Proof of Corollary 4.2

The update formulas in linear two-timescale algorithms are as follows:

$$\theta_{k+1} = \theta_k + \alpha_k[h_1(\theta_k, \omega_k) + M^{(1)}_{k+1}], \tag{26}$$

$$\omega_{k+1} = \omega_k + \alpha_k[h_2(\theta_k, \omega_k) + M^{(2)}_{k+1}]. \tag{27}$$

where $\alpha_k, \beta_k \in \mathbb{R}$ are stepsizes and $M^{(1)} \in \mathbb{R}^{d_1}$, $M^{(2)} \in \mathbb{R}^{d_2}$ denote noise. $h_1 : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \to \mathbb{R}^{d_1}$ and $h_2 : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \to \mathbb{R}^{d_2}$ have the form, respectively,

$$h_1(\theta, \omega) = v_1 - \Gamma_1\theta - W_1\omega, \tag{28}$$

$$h_2(\theta, \omega) = v_2 - \Gamma_2\theta - W_2\omega, \tag{29}$$

where $v_1 \in \mathbb{R}^{d_1}$, $v_2 \in \mathbb{R}^{d_2}$, $\Gamma_1 \in \mathbb{R}^{d_1 \times d_1}$, $\Gamma_2 \in \mathbb{R}^{d_2 \times d_1}$, $W_1 \in \mathbb{R}^{d_1 \times d_2}$ and $W_2 \in \mathbb{R}^{d_2 \times d_2}$. d_1 and d_2 are the dimensions of vectors θ and ω, respectively.

For Theorem 3 in (Dalal et al., 2020), the theorem still holds even when $d||1$ is not equal to d_2. For the VMTD algorithm, d_2 is equal to 1. (Dalal et al., 2020) presents the matrix assumption, step size assumption, and defines sparse projection.

Assumption A.1. (Matrix Assumption). W_2 and $X_1 = \Gamma_1 - W_1 W_2^{-1}\Gamma_2$ are positive definite(not necessarily symmetric).

Assumption A.2. (Step Size Assumption). $\alpha_k = (k+1)^{-\alpha}$ and $\beta_k = (k+1)^{-\beta}$, where $1 > \alpha > \beta > 0$.

Definition A.3. (Sparse Projection). For $R > 0$, let $\Pi_R(x) = \min\{1, R/||x||\}$. x be the projection into the ball with redius R around the origin. The sparse projection operator

$$\Pi_{n,R} = \begin{cases} \Pi_R, & \text{if } k = n^n - 1 \text{ for some } n \in \mathbb{Z}_{>0}, \\ I, & \text{otherwise.} \end{cases}$$

We call it sparse as it projects only on specific indices that are exponentially far apart.

Pick an arbitrary $p > 1$. Fix some constant $R^{\theta}_{\text{proj}} > 0$ and $R^{\omega}_{\text{proj}} > 0$ for the radius of the projection ball. Further, let

$$\theta^* = X_1^{-1} b_1, \omega^* = W_2^{-1}(v_2 - \Gamma_2\theta^*)$$

with $b_1 = v_1 - W_1 W_2^{-1} v_2$. The formula for the sparse projection update in linear two-timescale algorithms is as follows:

$$\theta'_{k+1} = \Pi_{k+1, R^{\theta}_{\text{proj}}}(\theta'_k + \alpha_k[h_1(\theta'_k, \omega'_k) + M^{(1')}_{k+1}]), \tag{30}$$

$$\omega'_{k+1} = \Pi_{k+1, R^{\omega}_{\text{proj}}}(\omega'_k + \beta_k[h_2(\theta'_k, \omega'_k) + M^{(2')}_{k+1}]). \tag{31}$$

Proof. As long as the VMTD algorithm satisfies Assumption A.1, the convergence speed of the VMTD algorithm can be obtained.

VMTD's update rule is

$$\theta_{k+1} = \theta_k + \alpha_k(\delta_k - \omega_k)\phi_k.$$

$$\omega_{k+1} = \omega_k + \beta_k(\delta_k - \omega_k).$$

Thus, $h_1(\theta, \omega) = \text{Cov}(r, \phi) - \text{Cov}(\phi, \phi - \gamma\phi')\theta$, $h_2(\theta, \omega) = \mathbb{E}[r] + \mathbb{E}[\gamma\phi'^{\top} - \phi^{\top}]\theta - \omega$, $\Gamma_1 = \text{Cov}(\phi, \phi - \gamma\phi')$, $W_1 = 0$ and $\Gamma_2 = -\mathbb{E}[\gamma\phi'^{\top} - \phi^{\top}]$, $W_2 = 1$, $v_2 = \mathbb{E}[r]$. Additionally, $X_1 = \Gamma_1 - W_1 W_2^{-1}\Gamma_2 = \text{Cov}(\phi, \phi - \gamma\phi')$. It can be deduced from the proof 4 that X_1 is a positive definite matrix. The VMTD algorithm satisfies the Assumption A.1. By the proof 4, Definition 1 in (Dalal et al., 2020) is satisfied. We can apply the Theorem 3 in (Dalal et al., 2020) to get the Corollary 4.2.

□

A.2. Proof of Theorem 4.3

Proof. The proof is similar to that given by (Sutton et al., 2009) for TDC, but it is based on multi-time-scale stochastic approximation.

For the VMTDC algorithm, a new one-step linear TD solution is defined as:

$$0 = \mathbb{E}[(\phi - \gamma\phi' - \mathbb{E}[\phi - \gamma\phi'])\phi^\top]\mathbb{E}[\phi\phi^\top]^{-1}\mathbb{E}[(\delta - \mathbb{E}[\delta])\phi] = A^\top C^{-1}(-A\theta + b).$$

The matrix $A^\top C^{-1} A$ is positive definite. Thus, the VMTD's solution is $\theta_{\text{VMTDC}} = \theta_{\text{VMTD}} = A^{-1}b$.

First, note that recursion (11) and (12) can be rewritten as, respectively,

$$\theta_{k+1} \leftarrow \theta_k + \zeta_k x(k),$$

$$u_{k+1} \leftarrow u_k + \beta_k y(k),$$

where

$$x(k) = \frac{\alpha_k}{\zeta_k}[(\delta_k - \omega_k)\phi_k - \gamma\phi'_k(\phi_k^\top u_k)],$$

$$y(k) = \frac{\zeta_k}{\beta_k}[\delta_k - \omega_k - \phi_k^\top u_k]\phi_k.$$

Recursion (11) can also be rewritten as

$$\theta_{k+1} \leftarrow \theta_k + \beta_k z(k),$$

where

$$z(k) = \frac{\alpha_k}{\beta_k}[(\delta_k - \omega_k)\phi_k - \gamma\phi'_k(\phi_k^\top u_k)],$$

Due to the settings of step-size schedule $\alpha_k = o(\zeta_k)$, $\zeta_k = o(\beta_k)$, $x(k) \to 0$, $y(k) \to 0$, $z(k) \to 0$ almost surely as $k \to 0$. That is that the increments in iteration (13) are uniformly larger than those in (12) and the increments in iteration (12) are uniformly larger than those in (11), thus (13) is the fastest recursion, (12) is the second fast recursion and (11) is the slower recursion. Along the fastest time scale, iterations of (11), (12) and (13) are associated to ODEs system as follows:

$$\dot{\theta}(t) = 0, \tag{32}$$

$$\dot{u}(t) = 0, \tag{33}$$

$$\dot{\omega}(t) = \mathbb{E}[\delta_t | u(t), \theta(t)] - \omega(t). \tag{34}$$

Based on the ODE (32) and (33), both $\theta(t) \equiv \theta$ and $u(t) \equiv u$ when viewed from the fastest timescale. By the Hirsch lemma (Hirsch, 1989), it follows that $||\theta_k - \theta|| \to 0$ a.s. as $k \to \infty$ for some θ that depends on the initial condition θ_0 of recursion (11) and $||u_k - u|| \to 0$ a.s. as $k \to \infty$ for some u that depends on the initial condition u_0 of recursion (12). Thus, the ODE pair (32)-(refomegavmtdcFastest) can be written as

$$\dot{\omega}(t) = \mathbb{E}[\delta_t | u, \theta] - \omega(t). \tag{35}$$

Consider the function $h(\omega) = \mathbb{E}[\delta | \theta, u] - \omega$, i.e., the driving vector field of the ODE (35). It is easy to find that the function h is Lipschitz with coefficient -1. Let $h_\infty(\cdot)$ be the function defined by $h_\infty(\omega) = \lim_{r \to \infty} \frac{h(r\omega)}{r}$. Then $h_\infty(\omega) = -\omega$, is well-defined. For (35), $\omega^* = \mathbb{E}[\delta | \theta, u]$ is the unique globally asymptotically stable equilibrium. For the ODE

$$\dot{\omega}(t) = h_\infty(\omega(t)) = -\omega(t), \tag{36}$$

apply $\vec{V}(\omega) = (-\omega)^\top(-\omega)/2$ as its associated strict Liapunov function. Then, the origin of (36) is a globally asymptotically stable equilibrium.

Consider now the recursion (13). Let $M_{k+1} = (\delta_k - \omega_k) - \mathbb{E}[(\delta_k - \omega_k)|\mathcal{F}(k)]$, where $\mathcal{F}(k) = \sigma(\omega_l, u_l, \theta_l, l \leq k; \phi_s, \phi'_s, r_s, s < k)$, $k \geq 1$ are the sigma fields generated by $\omega_0, u_0, \theta_0, \omega_{l+1}, u_{l+1}, \theta_{l+1}, \phi_l, \phi'_l$, $0 \leq l < k$. It is easy to verify that $M_{k+1}, k \geq 0$ are integrable random variables that satisfy $\mathbb{E}[M_{k+1}|\mathcal{F}(k)] = 0, \forall k \geq 0$. Because ϕ_k, r_k, and ϕ'_k have uniformly bounded second moments, it can be seen that for some constant $c_1 > 0, \forall k \geq 0$,

$$\mathbb{E}[||M_{k+1}||^2|\mathcal{F}(k)] \leq c_1(1 + ||\omega_k||^2 + ||u_k||^2 + ||\theta_k||^2).$$

Now Assumptions (A1) and (A2) of (Borkar & Meyn, 2000) are verified. Furthermore, Assumptions (TS) of (Borkar & Meyn, 2000) is satisfied by our conditions on the step-size sequences α_k,ζ_k, β_k. Thus, by Theorem 2.2 of (Borkar & Meyn, 2000) we obtain that $||\omega_k - \omega^*|| \to 0$ almost surely as $k \to \infty$.

Consider now the second time scale recursion (12). Based on the above analysis, (12) can be rewritten as

$$\dot{\theta}(t) = 0, \tag{37}$$

$$\dot{u}(t) = \mathbb{E}[(\delta_t - \mathbb{E}[\delta_t|u(t), \theta(t)])\phi_t|\theta(t)] - Cu(t). \tag{38}$$

The ODE (37) suggests that $\theta(t) \equiv \theta$ (i.e., a time invariant parameter) when viewed from the second fast timescale. By the Hirsch lemma (Hirsch, 1989), it follows that $||\theta_k - \theta|| \to 0$ a.s. as $k \to \infty$ for some θ that depends on the initial condition θ_0 of recursion (11).

Consider now the recursion (12). Let $N_{k+1} = ((\delta_k - \mathbb{E}[\delta_k]) - \phi_k\phi_k^\top u_k) - \mathbb{E}[((\delta_k - \mathbb{E}[\delta_k]) - \phi_k\phi_k^\top u_k)|\mathcal{I}(k)]$, where $\mathcal{I}(k) = \sigma(u_l, \theta_l, l \leq k; \phi_s, \phi'_s, r_s, s < k)$, $k \geq 1$ are the sigma fields generated by $u_0, \theta_0, u_{l+1}, \theta_{l+1}, \phi_l, \phi'_l$, $0 \leq l < k$. It is easy to verify that $N_{k+1}, k \geq 0$ are integrable random variables that satisfy $\mathbb{E}[N_{k+1}|\mathcal{I}(k)] = 0, \forall k \geq 0$. Because ϕ_k, r_k, and ϕ'_k have uniformly bounded second moments, it can be seen that for some constant $c_2 > 0, \forall k \geq 0$,

$$\mathbb{E}[||N_{k+1}||^2|\mathcal{I}(k)] \leq c_2(1 + ||u_k||^2 + ||\theta_k||^2).$$

Because $\theta(t) \equiv \theta$ from (37), the ODE pair (37)-(38) can be written as

$$\dot{u}(t) = \mathbb{E}[(\delta_t - \mathbb{E}[\delta_t|\theta])\phi_t|\theta] - Cu(t). \tag{39}$$

Now consider the function $h(u) = \mathbb{E}[\delta_t - \mathbb{E}[\delta_t|\theta]|\theta] - Cu$, i.e., the driving vector field of the ODE (39). For (39), $u^* = C^{-1}\mathbb{E}[(\delta - \mathbb{E}[\delta|\theta])\phi|\theta]$ is the unique globally asymptotically stable equilibrium. Let $h_\infty(u) = -Cu$. For the ODE

$$\dot{u}(t) = h_\infty(u(t)) = -Cu(t), \tag{40}$$

the origin of (40) is a globally asymptotically stable equilibrium because C is a positive definite matrix (because it is nonnegative definite and nonsingular). Now Assumptions (A1) and (A2) of (Borkar & Meyn, 2000) are verified. Furthermore, Assumptions (TS) of (Borkar & Meyn, 2000) is satisfied by our conditions on the step-size sequences α_k,ζ_k, β_k. Thus, by Theorem 2.2 of (Borkar & Meyn, 2000) we obtain that $||u_k - u^*|| \to 0$ almost surely as $k \to \infty$.

Consider now the slower timescale recursion (11). In the light of the above, (11) can be rewritten as

$$\theta_{k+1} \leftarrow \theta_k + \alpha_k(\delta_k - \mathbb{E}[\delta_k|\theta_k])\phi_k - \alpha_k\gamma\phi'_k(\phi_k^\top C^{-1}\mathbb{E}[(\delta_k - \mathbb{E}[\delta_k|\theta_k])\phi|\theta_k]). \tag{41}$$

Let $\mathcal{G}(k) = \sigma(\theta_l, l \leq k; \phi_s, \phi'_s, r_s, s < k)$, $k \geq 1$ be the sigma fields generated by $\theta_0, \theta_{l+1}, \phi_l, \phi'_l$, $0 \leq l < k$. Let

$$\begin{aligned} Z_{k+1} &= ((\delta_k - \mathbb{E}[\delta_k|\theta_k])\phi_k - \gamma\phi'_k\phi_k^\top C^{-1}\mathbb{E}[(\delta_k - \mathbb{E}[\delta_k|\theta_k])\phi|\theta_k]) \\ &\quad -\mathbb{E}[((\delta_k - \mathbb{E}[\delta_k|\theta_k])\phi_k - \gamma\phi'_k\phi_k^\top C^{-1}\mathbb{E}[(\delta_k - \mathbb{E}[\delta_k|\theta_k])\phi|\theta_k])|\mathcal{G}(k)] \\ &= ((\delta_k - \mathbb{E}[\delta_k|\theta_k])\phi_k - \gamma\phi'_k\phi_k^\top C^{-1}\mathbb{E}[(\delta_k - \mathbb{E}[\delta_k|\theta_k])\phi|\theta_k]) \\ &\quad -\mathbb{E}[(\delta_k - \mathbb{E}[\delta_k|\theta_k])\phi_k|\theta_k] - \gamma\mathbb{E}[\phi'\phi^\top]C^{-1}\mathbb{E}[(\delta_k - \mathbb{E}[\delta_k|\theta_k])\phi_k|\theta_k]. \end{aligned}$$

It is easy to see that Z_k, $k \geq 0$ are integrable random variables and $\mathbb{E}[Z_{k+1}|\mathcal{G}(k)] = 0, \forall k \geq 0$. Further,

$$\mathbb{E}[||Z_{k+1}||^2|\mathcal{G}(k)] \leq c_3(1 + ||\theta_k||^2), k \geq 0$$

for some constant $c_3 \geq 0$, again beacuse ϕ_k, r_k, and ϕ'_k have uniformly bounded second moments, it can be seen that for some constant.

Consider now the following ODE associated with (11):

$$\dot{\theta}(t) = (I - \mathbb{E}[\gamma\phi'\phi^\top]C^{-1})\mathbb{E}[(\delta - \mathbb{E}[\delta|\theta(t)])\phi|\theta(t)]. \tag{42}$$

Algorithm 2 VMTDC algorithm with linear function approximation in the off-policy setting

```
Input: $\theta_0$, $u_0$, $\omega_0$, $\gamma$, learning rate $\alpha_t$, $\zeta_t$ and $\beta_t$, behavior policy $\mu$ and target policy $\pi$
repeat
  For any episode, initialize $\theta_0$ arbitrarily, $u_t$ and $\omega_0$ to 0, $\gamma \in (0,1]$, and $\alpha_t$, $\zeta_t$ and $\beta_t$ are constant.
  Output: $\theta^*$.
  for $t = 0$ to $T-1$ do
    Take $A_t$ from $S_t$ according to $\mu$, and arrive at $S_{t+1}$
    Observe sample ($S_t$,$R_{t+1}$,$S_{t+1}$) at time step $t$ (with their corresponding state feature vectors)
    $\delta_t = R_{t+1} + \gamma\theta_t^\top \phi_{t+1} - \theta_t^\top \phi_t$
    $\rho_t \leftarrow \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)}$
    $\theta_{t+1} \leftarrow \theta_t + \alpha_t[\rho_t(\delta_t - \omega_t)\phi_t - \gamma\phi_{t+1}(\phi_t^\top u_t)]$
    $u_{t+1} \leftarrow u_t + \zeta_t[\rho_t(\delta_t - \omega_t) - \phi_t^\top u_t]\phi_t$
    $\omega_{t+1} \leftarrow \omega_t + \beta_t\rho_t(\delta_t - \omega_t)$
    $S_t = S_{t+1}$
  end for
until terminal episode
```

Let

$$\begin{array}{rcl}\vec{h}(\theta(t)) & = & (I - \mathbb{E}[\gamma\phi'\phi^\top]C^{-1})\mathbb{E}[(\delta - \mathbb{E}[\delta|\theta(t)])\phi|\theta(t)] \\ & = & (C - \mathbb{E}[\gamma\phi'\phi^\top])C^{-1}\mathbb{E}[(\delta - \mathbb{E}[\delta|\theta(t)])\phi|\theta(t)] \\ & = & (\mathbb{E}[\phi\phi^\top] - \mathbb{E}[\gamma\phi'\phi^\top])C^{-1}\mathbb{E}[(\delta - \mathbb{E}[\delta|\theta(t)])\phi|\theta(t)] \\ & = & A^\top C^{-1}(-A\theta(t) + b),\end{array}$$

because $\mathbb{E}[(\delta - \mathbb{E}[\delta|\theta(t)])\phi|\theta(t)] = -A\theta(t) + b$, where $A = \mathrm{Cov}(\phi, \phi - \gamma\phi')$, $b = \mathrm{Cov}(r, \phi)$, and $C = \mathbb{E}[\phi\phi^\top]$

Therefore, $\theta^* = A^{-1}b$ can be seen to be the unique globally asymptotically stable equilibrium for ODE (42). Let $\vec{h}_\infty(\theta) = \lim_{r\to\infty}\frac{\vec{h}(r\theta)}{r}$. Then $\vec{h}_\infty(\theta) = -A^\top C^{-1}A\theta$ is well-defined. Consider now the ODE

$$\dot{\theta}(t) = -A^\top C^{-1}A\theta(t). \tag{43}$$

Because C^{-1} is positive definite and A has full rank (as it is nonsingular by assumption), the matrix $A^\top C^{-1}A$ is also positive definite. The ODE (43) has the origin as its unique globally asymptotically stable equilibrium. Thus, the assumption (A1) and (A2) are verified.

The proof is given above. In the fastest time scale, the parameter w converges to $\mathbb{E}[\delta|u_k, \theta_k]$. In the second fast time scale, the parameter u converges to $C^{-1}\mathbb{E}[(\delta - \mathbb{E}[\delta|\theta_k])\phi|\theta_k]$. In the slower time scale, the parameter θ converges to $A^{-1}b$. □

B. Experimental details

The feature matrices corresponding to three random walks are shown below respectively:

$$\Phi_{tabular} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Phi_{inverted} = \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}$$

$$\Phi_{dependent} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & 1 \end{bmatrix}$$

Three random walk experiments: the α values for all algorithms are in the range of $\{0.008, 0.015, 0.03, 0.06, 0.12, 0.25, 0.5\}$. For the TDC algorithm, the range of the ratio $\frac{\zeta}{\alpha}$ is $\{\frac{1}{512}, \frac{1}{256}, \frac{1}{128}, \frac{1}{64}, \frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2\}$. For the VMTD algorithm, the range of the ratio $\frac{\beta}{\alpha}$ is $\{\frac{1}{512}, \frac{1}{256}, \frac{1}{128}, \frac{1}{64}, \frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2\}$. It can be observed from the update formula of VMTDC that when ζ takes a very small value, the VMTDC update tends to be similar to VMTD update. Similarly, when β takes a very small value, the VMTDC update tends to be similar to TDC update. Through experiments, it was found that setting ζ to a small value makes VMTDC updates approach VMTD updates, resulting in better performance. Therefore, for the VMTDC algorithm, the range of $\frac{\beta}{\alpha}$ ratio is $\{\frac{1}{512}, \frac{1}{256}, \frac{1}{128}, \frac{1}{64}, \frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2\}$, and the range of ζ is $\{0.1, 0.01, 0.001, 0.0001, 0.00001\}$. The learning curves in Figure 3 correspond to the optimal parameters.

The feature matrix of 7-state version of Baird's off-policy counterexample is defined as follow:

$$\Phi_{Counter} = \begin{bmatrix} 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

7-state version of Baird's off-policy counterexample: for TD algorithm, α is set to 0.1. For the TDC algorithm, the range of α is $\{0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$, and the range of ζ is $\{0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5\}$. For the VMTD algorithm, the range of α is $\{0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$, and the range of β is $\{0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5\}$. Through experiments, it was found that setting ζ to a small value makes VMTDC updates approach VMTD updates, resulting in better performance. Therefore, for the VMTDC algorithm, The range of values for α and β is the same as that of VMTD and the range of ζ is $\{0.1, 0.01, 0.001, 0.0001, 0.00001\}$. The learning curves in Figure 4 correspond to the optimal parameters. For all policy evaluation experiments, each experiment is independently run 100 times.

For the four control experiments: The learning rates for each algorithm in all experiments are shown in Table 3. For all control experiments, each experiment is independently run 50 times.

Table 3. Learning rates (lr) of four control experiments.

envs algorithms(lr)	Maze	Cliff walking	Mountain Car	Acrobot
Sarsa(α)	0.1	0.1	0.1	0.1
GQ(0)(α, ζ)	$0.1, 0.003$	$0.1, 0.004$	$0.1, 0.01$	$0.1, 0.01$
VMSarsa(α, β)	$0.1, 0.001$	0.1, 1e-4	0.1, 1e-4	0.1, 1e-4
VMGQ(0)(α, ζ, β)	$0.1, 0.001, 0.001$	0.1, 0.005, 1e-4	0.1, 5e-4, 1e-4	0.1, 5e-4, 1e-4
AC($lr_{\text{actor}}, lr_{\text{critic}}$)	$0.01, 0.1$	$0.01, 0.01$	$0.01, 0.05$	$0.01, 0.05$
Q-learning(α)	0.1	0.1	0.1	0.1
VMQ(α, β)	$0.1, 0.001$	0.1, 1e-4	0.1, 1e-4	0.1, 1e-4