# A Variance Minimization Approach to Off-policy Temporal-Difference Learning

**Anonymous submission**


## Abstract

In this paper, we introduce the concept of improving the performance of parametric Temporal-Difference (TD) learning algorithms by the Variance Minimization (VM) parameter, $\omega$, which is dynamically updated at each time step. Specifically, we incorporate the VM parameter into off-policy linear algorithms such as TDC and ETD, resulting in the Variance Minimization TDC (VMTDC) algorithm and the Variance Minimization ETD (VMETD) algorithm. In the two-state counterexample, we analyze the convergence speed of these algorithms by calculating the minimum eigenvalue of the key matrices and find that the VMTDC algorithm converges faster than TDC, while VMETD is more stable in convergence than ETD through the experiment.In controlled experiments, the VM algorithms demonstrate superior performance.


## 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 and Van Roy (1997) established the convergence of on-policy TD($\lambda$). However, "The deadly triad" consisting of off-policy learning, bootstrapping, and function approximation makes the stability a difficult problem (Sutton and Barto 2018). To solve this problem, convergent off-policy temporal difference learning algorithms are proposed, e.g., BR (Baird et al. 1995), GTD (Sutton, Maei, and Szepesvári 2008), GTD2 and TDC (Sutton et al. 2009), ETD (Sutton, Mahmood, and White 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 and Palhang 2015) and accelerated TD (ATD) (Pan, White, and White 2017). Hallak et al. (2016) introduced an new parameter to reduce variance for ETD. Zhang and Whiteson (2022) proposed truncated ETD with a lower variance. Variance Reduced TD with direct variance reduction technique (Johnson and Zhang 2013) is proposed by (Korda and 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 and 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, Li, and Liu 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 error. Is minimizing error the only option for value-based reinforcement learning?

For policy evaluation experiments, differences in objective functions may result in inconsistent fixed points. This inconsistency makes it difficult to uniformly compare the superiority of algorithms derived from different objective functions. However, for control experiments, since the choice of actions depends on the relative values of the Q values rather than their absolute values, the presence of solution bias is acceptable.

Based on this observation, we propose alternate objective functions instead of minimizing errors. We minimize Variance of Projected Bellman Error (VPBE) and derive Variance Minimization (VM) algorithms. These algorithms preserve the invariance of the optimal policy in the control environments, 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) Propose two off-policy variance minimization algorithms. (3) Proof of their convergence. (5) Experiments demonstrating the faster convergence speed of the proposed algorithms.

# Background

## Markov Decision Process

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 and 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 $\pi^*$ to maximize the expectation of a discounted cumulative rewards in a long period. State value function $V^\pi(s)$ for a stationary policy $\pi$ 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:

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

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

Tabular temporal difference (TD) learning (Sutton and 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 (the focus of this paper), kernel methods, decision trees, or neural networks, etc.

## On-policy and Off-policy

On-policy and off-policy algorithms are currently hot topics in research. Off-policy algorithms, in particular, present greater challenges due to the difficulty in ensuring their convergence, making them more complex to study. The main difference between the two lies in the fact that in on-policy algorithms, the behavior policy $\mu$ and the target policy $\pi$ are the same during the learning process. The algorithm directly generates data from the current policy and optimizes it. In off-policy algorithms, however, the behavior policy and the target policy are different. The algorithm uses data generated from the behavior policy to optimize the target policy, which leads to higher sample efficiency and complex stability issues.

Taking the TD(0) algorithm as an example can help understand the different performances of on-policy and off-policy:

In the on-policy TD(0) algorithm, the behavior policy and the target policy are the same. The algorithm uses the data generated by the current policy to update its value estimates. Since the behavior policy and the target policy are consistent, the convergence of TD(0) is more assured. In each step of the update, the algorithm is based on the actual behavior of the current policy, which gradually leads the value function estimate to converge to the true value of the target policy.

The on-policy TD(0) update formula is

$$\boldsymbol{\theta}_{k+1} \quad \leftarrow \quad \boldsymbol{\theta}_k + \alpha_k \delta_k \boldsymbol{\phi}_k,$$

where $\delta_k = r_{k+1} + \gamma \boldsymbol{\theta}_k^\top \boldsymbol{\phi}_{k+1} - \boldsymbol{\theta}_k^\top \boldsymbol{\phi}_k$ and the key matrix $\mathbf{A}_{\text{on}}$ of on-policy TD(0) is

$$\mathbf{A}_{\text{on}} = \boldsymbol{\Phi}^\top \mathbf{D}_\pi (\mathbf{I} - \gamma \mathbf{P}_\pi) \boldsymbol{\Phi},$$

where $\boldsymbol{\Phi}$ is the $N \times n$ matrix with the $\phi(s)$ as its rows, and $\mathbf{D}_\pi$ is the $N \times N$ diagonal matrix with $\mathbf{d}_\pi$ on its diagonal. $\mathbf{d}_\pi$ is a vector, each component representing the steady-state distribution under $\pi$. $\mathbf{P}_\pi$ denote the $N \times N$ matrix of transition probabilities under $\pi$. And $\mathbf{P}_\pi^\top \mathbf{d}_\pi = \mathbf{d}_\pi$.

An $\boldsymbol{\Phi}^\top \mathrm{X} \boldsymbol{\Phi}$ matrix of this form will be positive definite whenever the matrix X is positive definite. Any matrix X is positive definite if and only if the symmetric matrix $\mathrm{S} = \mathrm{X} + \mathrm{X}^\top$ is positive definite. Any symmetric real matrix S is positive definite if the absolute values of its diagonal entries are greater than the sum of the absolute values of the corresponding off-diagonal entries(Sutton, Mahmood, and White 2016).

All components of the matrix $\mathbf{D}_\pi(\mathbf{I} - \gamma \mathbf{P}_\pi)$ are positive. The row sums of $\mathbf{D}_\pi(\mathbf{I} - \gamma \mathbf{P}_\pi)$ are positive. And The row sums of $\mathbf{D}_\pi(\mathbf{I} - \gamma \mathbf{P}_\pi)$ are

$$\begin{aligned} \mathbf{1}^\top \mathbf{D}_\pi (\mathbf{I} - \gamma \mathbf{P}_\pi) &= \mathbf{d}_\pi^\top (\mathbf{I} - \gamma \mathbf{P}_\pi) \\ &= \mathbf{d}_\pi^\top - \gamma \mathbf{d}_\pi^\top \mathbf{P}_\pi \\ &= \mathbf{d}_\pi^\top - \gamma \mathbf{d}_\pi^\top \\ &= (1 - \gamma) \mathbf{d}_\pi^\top, \end{aligned}$$

all components of which are positive. Thus, the key matrix and its $\mathbf{A}_{\text{on}}$ matrix are positive definite, and on-policy TD(0) is stable

The off-policy TD(0) update formula is

$$\boldsymbol{\theta}_{k+1} \quad \leftarrow \quad \boldsymbol{\theta}_k + \alpha_k \rho_k \delta_k \boldsymbol{\phi}_k,$$

where $\rho_k = \frac{\pi(A_k|S_k)}{\mu(A_k|S_k)}$, called importance sampling ratio, and the key matrix $\mathbf{A}_{\text{off}}$ of off-policy TD(0) is

$$\mathbf{A}_{\text{off}} = \boldsymbol{\Phi}^\top \mathbf{D}_\mu (\mathbf{I} - \gamma \mathbf{P}_\pi) \boldsymbol{\Phi}.$$

where $\mathbf{D}_\mu$ is the $N \times N$ diagonal matrix with $\mathbf{d}_\mu$ on its diagonal. $\mathbf{d}_\mu$ is a vector, each component representing the steady-state distribution under $\mu$

If the key matrix $\mathbf{A}$ in the algorithm is positive definite, then the algorithm is stable and converges. However, in the off-policy TD(0) algorithm, it cannot be guaranteed that $\mathbf{A}$ is a positive definite matrix. In the 2-state counterexample, $\mathbf{A}_{\text{off}} = -0.2$, which means that off-policy TD(0) cannot stably converge.

TDC and ETD are two well-known off-policy algorithms. The former is an off-policy algorithm derived from the objective function Mean Squared Projected Bellman error (MSPBE), while the latter employs a technique to transform the key matrix $\mathbf{A}$ in the original off-policy TD(0) from non-positive definite to positive definite, thereby ensuring the algorithm's convergence under off-policy conditions.

The MSPBE with importance sampling is

$$\begin{aligned} \text{MSPBE}(\boldsymbol{\theta}) &= ||\mathbf{V}_{\boldsymbol{\theta}} - \Pi \mathbf{T}^\pi \mathbf{V}_{\boldsymbol{\theta}}||_\mu^2 \\ &= ||\Pi(\mathbf{V}_{\boldsymbol{\theta}} - \mathbf{T}^\pi \mathbf{V}_{\boldsymbol{\theta}})||_\mu^2 \\ &= \mathbb{E}[\rho\delta\boldsymbol{\phi}]^\top \mathbb{E}[\boldsymbol{\phi}\boldsymbol{\phi}^\top]^{-1} \mathbb{E}[\rho\delta\boldsymbol{\phi}], \end{aligned}$$

where $\mathbf{V}_{\boldsymbol{\theta}}$ is viewed as vectors with one element for each state, the norm $||\boldsymbol{v}||_\mu^2 = \sum_s \mu(s)\boldsymbol{v}^2(s)$, $\mathbf{T}^\pi$, simplified to $\mathbf{T}$ in the following text, is Bellman operator and $\mathbf{\Pi} = \mathbf{\Phi}(\mathbf{\Phi}^\top \mathbf{D} \mathbf{\Phi})^{-1} \mathbf{\Phi}^\top \mathbf{D}$. The TDC update formula with importance sampling is

$$\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k + \alpha_k \rho_k [\delta_k \boldsymbol{\phi}_k - \gamma \boldsymbol{\phi}_{k+1} (\boldsymbol{\phi}_k^\top \boldsymbol{u}_k)],$$
$$\boldsymbol{u}_{k+1} \leftarrow \boldsymbol{u}_k + \zeta_k [\rho_k \delta_k - \boldsymbol{\phi}_k^\top \boldsymbol{u}_k] \boldsymbol{\phi}_k.$$

The key matrix $\mathbf{A}_{\text{TDC}} = \mathbf{A}_{\text{off}}^\top \mathbf{C}^{-1} \mathbf{A}_{\text{off}}$, where $\mathbf{C} = \mathbb{E}[\boldsymbol{\phi}\boldsymbol{\phi}^\top]$. In the 2-state counterexample, $\mathbf{A}_{\text{TDC}} = 0.016$, which means that TDC can stably converge.

The ETD update formula is

$$\begin{aligned} F_k &\leftarrow \gamma \rho_{k-1} F_{k-1} + 1, \\ \boldsymbol{\theta}_{k+1} &\leftarrow \boldsymbol{\theta}_k + \alpha_k F_k \rho_k \delta_k \boldsymbol{\phi}_k, \end{aligned} \tag{2}$$

where $F_t$ is a scalar variable and $F_0 = 1$. The key matrix $\mathbf{A}_{\text{ETD}} = \mathbf{\Phi}^\top \mathbf{F} (\mathbf{I} - \gamma \mathbf{P}_\pi) \mathbf{\Phi}$, where $\mathbf{F}$ is a diagonal matrix with diagonal elements $f(s) \doteq d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu[F_k | S_k = s]$, which we assume exists. The vector $\mathbf{f} \in \mathbb{R}^N$ with components $[\mathbf{f}]_s \doteq f(s)$ can be written as

$$\begin{aligned} \mathbf{f} &= \mathbf{d}_\mu + \gamma \mathbf{P}_\pi^\top \mathbf{d}_\mu + (\gamma \mathbf{P}_\pi^\top)^2 \mathbf{d}_\mu + \dots \\ &= (\mathbf{I} - \gamma \mathbf{P}_\pi^\top)^{-1} \mathbf{d}_\mu. \end{aligned}$$

. The row sums of $\mathbf{F}(\mathbf{I} - \gamma \mathbf{P}_\pi)$ are

$$\begin{aligned} \mathbf{1}^\top \mathbf{F} (\mathbf{I} - \gamma \mathbf{P}_\pi) &= \mathbf{f}^\top (\mathbf{I} - \gamma \mathbf{P}_\pi) \\ &= \mathbf{d}_\mu^\top (\mathbf{I} - \gamma \mathbf{P}_\pi)^{-1} (\mathbf{I} - \gamma \mathbf{P}_\pi) \\ &= \mathbf{d}_\mu^\top, \end{aligned}$$

and in the 2-state counterexample, $\mathbf{A}_{\text{ETD}} = 3.4$, which means that ETD can stably converge.

The convergence rate of the algorithm is related to the matrix $\mathbf{A}$. The larger the minimum eigenvalue of $\mathbf{A}$, the faster the convergence rate. In the 2-state case, the minimum eigenvalue of the matrix $\mathbf{A}$ in ETD is the largest, so it converges the fastest. Based on this theorem, can we derive an algorithm with a larger minimum eigenvalue for matrix $\mathbf{A}$.

Algorithm 1: VMTDC algorithm with linear function approximation in the off-policy setting

**Input:** $\boldsymbol{\theta}_0$, $\boldsymbol{u}_0$, $\omega_0$, $\gamma$, learning rate $\alpha_t$, $\zeta_t$ and $\beta_t$, behavior policy $\mu$ and target policy $\pi$
**repeat**
  For any episode, initialize $\boldsymbol{\theta}_0$ arbitrarily, $\boldsymbol{u}_0$ and $\omega_0$ to 0, $\gamma \in (0, 1]$, and $\alpha_t$, $\zeta_t$ and $\beta_t$ are constant.
  **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)
    $\delta_t = R_{t+1} + \gamma \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_{t+1} - \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_t$
    $\rho_t \leftarrow \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)}$
    $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t + \alpha_t[(\rho_t \delta_t - \omega_t)\boldsymbol{\phi}_t - \gamma \rho_t \boldsymbol{\phi}_{t+1} (\boldsymbol{\phi}_t^\top \boldsymbol{u}_t)]$
    $\boldsymbol{u}_{t+1} \leftarrow \boldsymbol{u}_t + \zeta_t[(\rho_t \delta_t - \omega_t) - \boldsymbol{\phi}_t^\top \boldsymbol{u}_t]\boldsymbol{\phi}_t$
    $\omega_{t+1} \leftarrow \omega_t + \beta_t(\rho_t \delta_t - \omega_t)$
    $S_t = S_{t+1}$
  **end for**
**until** terminal episode

## Variance Minimization Algorithms

To derive an algorithm with a larger minimum eigenvalue for matrix $\mathbf{A}$, it is necessary to propose new objective functions. The mentioned objective functions in the Introduction are all forms of error. Is minimizing error the only option for value-based reinforcement learning? Based on this observation, we propose alternative objective functions instead of minimizing errors. We minimize the Variance of Projected Bellman Error (VPBE) and derive the VMTDC algorithm. This idea is then innovatively applied to ETD, resulting in the VMETD algorithm.

### Variance Minimization TDC Learning: VMTDC

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

$$\begin{aligned} \text{VPBE}(\boldsymbol{\theta}) &= \mathbb{E}[(\delta - \mathbb{E}[\delta])\boldsymbol{\phi}]^\top \mathbb{E}[\boldsymbol{\phi}\boldsymbol{\phi}^\top]^{-1} \\ &\quad \mathbb{E}[(\delta - \mathbb{E}[\delta])\boldsymbol{\phi}] \\ &= (\mathbf{\Phi}^\top \mathbf{D}(\mathbf{W}_{\boldsymbol{\theta}} + \mathbf{T}\mathbf{V}_{\boldsymbol{\theta}} - \mathbf{V}_{\boldsymbol{\theta}}))^\top (\mathbf{\Phi}^\top \mathbf{D} \mathbf{\Phi})^{-1} \\ &\quad \mathbf{\Phi}^\top \mathbf{D}(\mathbf{W}_{\boldsymbol{\theta}} + \mathbf{T}\mathbf{V}_{\boldsymbol{\theta}} - \mathbf{V}_{\boldsymbol{\theta}}) \\ &= (\mathbf{W}_{\boldsymbol{\theta}} + \mathbf{T}\mathbf{V}_{\boldsymbol{\theta}} - \mathbf{V}_{\boldsymbol{\theta}})^\top \mathbf{D}^\top \mathbf{\Phi} (\mathbf{\Phi}^\top \mathbf{D} \mathbf{\Phi})^{-1} \\ &\quad \mathbf{\Phi}^\top \mathbf{D}(\mathbf{W}_{\boldsymbol{\theta}} + \mathbf{T}\mathbf{V}_{\boldsymbol{\theta}} - \mathbf{V}_{\boldsymbol{\theta}}) \\ &= (\mathbf{W}_{\boldsymbol{\theta}} + \mathbf{T}\mathbf{V}_{\boldsymbol{\theta}} - \mathbf{V}_{\boldsymbol{\theta}})^\top \Pi^\top \mathbf{D} \Pi \\ &\quad (\mathbf{W}_{\boldsymbol{\theta}} + \mathbf{T}\mathbf{V}_{\boldsymbol{\theta}} - \mathbf{V}_{\boldsymbol{\theta}}) \\ &= (\Pi(\mathbf{V}_{\boldsymbol{\theta}} - \mathbf{T}\mathbf{V}_{\boldsymbol{\theta}} - \mathbf{W}_{\boldsymbol{\theta}}))^\top \mathbf{D} \\ &\quad (\Pi(\mathbf{V}_{\boldsymbol{\theta}} - \mathbf{T}\mathbf{V}_{\boldsymbol{\theta}} - \mathbf{W}_{\boldsymbol{\theta}})) \\ &= ||\Pi(\mathbf{V}_{\boldsymbol{\theta}} - \mathbf{T}\mathbf{V}_{\boldsymbol{\theta}} - \mathbf{W}_{\boldsymbol{\theta}})||_\mu^2 \\ &= ||\Pi(\mathbf{V}_{\boldsymbol{\theta}} - \mathbf{T}\mathbf{V}_{\boldsymbol{\theta}}) - \Pi \mathbf{W}_{\boldsymbol{\theta}}||_\mu^2 \\ &= \mathbb{E}[(\delta - \omega)\boldsymbol{\phi}]^\top \mathbb{E}[\boldsymbol{\phi}\boldsymbol{\phi}^\top]^{-1} \mathbb{E}[(\delta - \omega)\boldsymbol{\phi}] \end{aligned} \tag{3}$$

(4)

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

**Input:** $\boldsymbol{\theta}_0$, $F_0$, $\omega_0$, $\gamma$, learning rate $\alpha_t$, $\zeta_t$ and $\beta_t$, behavior policy $\mu$ and target policy $\pi$
**repeat**
For any episode, initialize $\boldsymbol{\theta}_0$ arbitrarily, $F_0$ to 1 and $\omega_0$ to 0, $\gamma \in (0, 1]$, and $\alpha_t$, $\zeta_t$ and $\beta_t$ are constant.
**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)
$\delta_t = R_{t+1} + \gamma \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_{t+1} - \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_t$
$\rho_t \leftarrow \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)}$
$F_t \leftarrow \gamma \rho_t F_{t-1} + 1$
$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t + \alpha_t(F_t \rho_t \delta_t - \omega_t)\boldsymbol{\phi}_t$
$\omega_{t+1} \leftarrow \omega_t + \beta_t(F_t \rho_t \delta_t - \omega_t)$
$S_t = S_{t+1}$
**end for**
**until** terminal episode

where $\mathbf{W}_{\boldsymbol{\theta}}$ is viewed as vectors with every element being equal to $||\mathbf{V}_{\boldsymbol{\theta}} - \mathbf{T}\mathbf{V}_{\boldsymbol{\theta}}||^2_\mu$ and $\omega$ is used to approximate $\mathbb{E}[\delta]$, i.e., $\omega \doteq \mathbb{E}[\delta]$.

The gradient of the (3) with respect to $\theta$ is

$$\begin{aligned}
-\tfrac{1}{2}\nabla \text{VPBE}(\boldsymbol{\theta}) &= -\mathbb{E}\Big[\Big((\gamma\boldsymbol{\phi}' - \boldsymbol{\phi}) - \mathbb{E}[(\gamma\boldsymbol{\phi}' - \boldsymbol{\phi})]\Big)\boldsymbol{\phi}^\top\Big] \\
&\quad \mathbb{E}[\boldsymbol{\phi}\boldsymbol{\phi}^\top]^{-1}\mathbb{E}[(\delta - \mathbb{E}[\delta])\boldsymbol{\phi}] \\
&= \mathbb{E}\Big[\Big((\boldsymbol{\phi} - \gamma\boldsymbol{\phi}') - \mathbb{E}[(\boldsymbol{\phi} - \gamma\boldsymbol{\phi}')]\Big)\boldsymbol{\phi}^\top\Big] \\
&\quad \mathbb{E}[\boldsymbol{\phi}\boldsymbol{\phi}^\top]^{-1} \\
&\quad \mathbb{E}\Big[\Big(r + \gamma\boldsymbol{\phi}'^\top\boldsymbol{\theta} - \boldsymbol{\phi}^\top\boldsymbol{\theta} \\
&\qquad -\mathbb{E}[r + \gamma\boldsymbol{\phi}'^\top\boldsymbol{\theta} - \boldsymbol{\phi}^\top\boldsymbol{\theta}]\Big)\boldsymbol{\phi}\Big] \\
&= \mathbf{A}^\top\mathbf{C}^{-1}(-\mathbf{A}\boldsymbol{\theta} + \mathbf{b})
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{A} &= \mathbb{E}\Big[\Big((\boldsymbol{\phi} - \gamma\boldsymbol{\phi}') - \mathbb{E}[(\boldsymbol{\phi} - \gamma\boldsymbol{\phi}')]\Big)\boldsymbol{\phi}^\top\Big] \\
&= \mathbb{E}[(\boldsymbol{\phi} - \gamma\boldsymbol{\phi}')\boldsymbol{\phi}^\top] - \mathbb{E}[\boldsymbol{\phi} - \gamma\boldsymbol{\phi}']\mathbb{E}[\boldsymbol{\phi}^\top] \\
&= \text{Cov}(\boldsymbol{\phi}, \boldsymbol{\phi} - \gamma\boldsymbol{\phi}'), \\
\mathbf{C} &= \mathbb{E}[\boldsymbol{\phi}\boldsymbol{\phi}^\top], \\
\mathbf{b} &= \mathbb{E}(r - \mathbb{E}[r])\boldsymbol{\phi} \\
&= \mathbb{E}[r\boldsymbol{\phi}] - \mathbb{E}[r]\mathbb{E}[\boldsymbol{\phi}] \\
&= \text{Cov}(r, \boldsymbol{\phi}),
\end{aligned}$$

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

In the process of computing the gradient of the (4) with respect to $\theta$, $\omega$ is treated as a constant. So, the derivation process of the VMTDC algorithm is the same as that of the TDC algorithm, the only difference is that the original $\delta$ is replaced by $\delta - \omega$. Therefore, we can easily get the updated formula of VMTDC, as follows:

$$\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k + \alpha_k[(\delta_k - \omega_k)\boldsymbol{\phi}_k - \gamma\boldsymbol{\phi}_{k+1}(\boldsymbol{\phi}_k^\top\boldsymbol{u}_k)], \quad (5)$$

$$\boldsymbol{u}_{k+1} \leftarrow \boldsymbol{u}_k + \zeta_k[\delta_k - \omega_k - \boldsymbol{\phi}_k^\top\boldsymbol{u}_k]\boldsymbol{\phi}_k, \quad (6)$$

and

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

The VMTDC algorithm (5) is derived to work with a given set of sub-samples—in the form of triples $(S_k, R_k, S'_k)$ that match transitions from both the behavior and target policies. What if we wanted to use all the data? The data is generated according to the behavior policy $\pi_b$, while our objective is to learn about the target policy $\pi$. We should use importance-sampling. The VPBE with importance sampling is:

$$\begin{aligned}
\text{VPBE}(\boldsymbol{\theta}) &= \mathbb{E}[(\rho\delta - \mathbb{E}[\rho\delta])\boldsymbol{\phi}]^\top\mathbb{E}[\boldsymbol{\phi}\boldsymbol{\phi}^\top]^{-1} \\
&\quad \mathbb{E}[(\rho\delta - \mathbb{E}[\rho\delta])\boldsymbol{\phi}],
\end{aligned} \quad (8)$$

Following the linear VMTDC derivation, we get the following algorithm (linear VMTDC algorithm based on importance weighting scenario):

$$\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k + \alpha_k[(\rho_k\delta_k - \omega_k)\boldsymbol{\phi}_k - \gamma\rho_k\boldsymbol{\phi}_{k+1}(\boldsymbol{\phi}_k^\top\boldsymbol{u}_k)], \quad (9)$$

$$\boldsymbol{u}_{k+1} \leftarrow \boldsymbol{u}_k + \zeta_k[(\rho_k\delta_k - \omega_k) - \boldsymbol{\phi}_k^\top\boldsymbol{u}_k]\boldsymbol{\phi}_k, \quad (10)$$

and

$$\omega_{k+1} \leftarrow \omega_k + \beta_k(\rho_k\delta_k - \omega_k), \quad (11)$$

The gradient of the (8) with respect to $\theta$ is

$$\begin{aligned}
-\tfrac{1}{2}\nabla \text{VPBE}(\boldsymbol{\theta}) &= \mathbb{E}\Big[\Big(\rho(\boldsymbol{\phi} - \gamma\boldsymbol{\phi}') - \mathbb{E}[\rho(\boldsymbol{\phi} - \gamma\boldsymbol{\phi}')]\Big)\boldsymbol{\phi}^\top\Big] \\
&\quad \mathbb{E}[\boldsymbol{\phi}\boldsymbol{\phi}^\top]^{-1} \\
&\quad \mathbb{E}\Big[\Big(\rho(r + \gamma\boldsymbol{\phi}'^\top\boldsymbol{\theta} - \boldsymbol{\phi}^\top\boldsymbol{\theta}) \\
&\qquad -\mathbb{E}[\rho(r + \gamma\boldsymbol{\phi}'^\top\boldsymbol{\theta} - \boldsymbol{\phi}^\top\boldsymbol{\theta})]\Big)\boldsymbol{\phi}\Big] \\
&= \mathbb{E}[\rho(\boldsymbol{\phi} - \gamma\boldsymbol{\phi}')\boldsymbol{\phi}^\top] - \mathbb{E}[\rho(\boldsymbol{\phi} - \gamma\boldsymbol{\phi}')]\mathbb{E}[\boldsymbol{\phi}^\top] \\
&\quad \mathbb{E}[\boldsymbol{\phi}\boldsymbol{\phi}^\top]^{-1} \\
&\quad \mathbb{E}\Big[\Big(\rho(r + \gamma\boldsymbol{\phi}'^\top\boldsymbol{\theta} - \boldsymbol{\phi}^\top\boldsymbol{\theta}) \\
&\qquad -\mathbb{E}[\rho(r + \gamma\boldsymbol{\phi}'^\top\boldsymbol{\theta} - \boldsymbol{\phi}^\top\boldsymbol{\theta})]\Big)\boldsymbol{\phi}\Big] \\
&= \mathbf{A}^\top\mathbf{C}^{-1}(-\mathbf{A}\boldsymbol{\theta} + \mathbf{b}),
\end{aligned}$$

where $\mathbf{A} = \boldsymbol{\Phi}^\top(\mathbf{D}_\mu - \mathbf{d}_\mu\mathbf{d}_\mu^\top)(\mathbf{I} - \gamma\mathbf{P}_\pi)\boldsymbol{\Phi}$, $\mathbf{b} = \boldsymbol{\Phi}^\top(\mathbf{D}_\mu - \mathbf{d}_\mu\mathbf{d}_\mu^\top)\mathbf{r}_\pi$ and $\mathbf{r}_\pi$ is viewed as vectors. In the 2-state counterexample, $\mathbf{A}_{\text{VMTDC}} = 0.025$, meaning that VMTDC can stably converge and converges faster than TDC.

## Variance Minimization ETD Learning: VMETD

Based on the off-policy TD algorithm, a scalar, $F$, is introduced to obtain the ETD algorithm, which ensures convergence under off-policy conditions. This paper further introduces a scalar, $\omega$, based on the ETD algorithm to obtain VMETD. VMETD by the following update:

$$\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k + \alpha_k(F_k\rho_k\delta_k - \omega_k)\boldsymbol{\phi}_k, \quad (12)$$

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

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

(12) can be rewritten as

$$\begin{aligned}
\boldsymbol{\theta}_{k+1} &\leftarrow \boldsymbol{\theta}_k + \alpha_k(F_k\rho_k\delta_k - \omega_k)\boldsymbol{\phi}_k - \alpha_k\omega_{k+1}\boldsymbol{\phi}_k \\
&= \boldsymbol{\theta}_k + \alpha_k(F_k\rho_k\delta_k - \mathbb{E}_\mu[F_k\rho_k\delta_k|\boldsymbol{\theta}_k])\boldsymbol{\phi}_k \\
&= \boldsymbol{\theta}_k + \alpha_k F_k\rho_k(R_{k+1} + \gamma\boldsymbol{\theta}_k^\top\boldsymbol{\phi}_{k+1} - \boldsymbol{\theta}_k^\top\boldsymbol{\phi}_k)\boldsymbol{\phi}_k \\
&\quad -\alpha_k\mathbb{E}_\mu[F_k\rho_k\delta_k]\boldsymbol{\phi}_k \\
&= \boldsymbol{\theta}_k + \alpha_k\{\underbrace{(F_k\rho_k R_{k+1} - \mathbb{E}_\mu[F_k\rho_k R_{k+1}])\boldsymbol{\phi}_k}_{\mathbf{b}_{\text{VMETD},k}} \\
&\quad \underbrace{-(F_k\rho_k\boldsymbol{\phi}_k(\boldsymbol{\phi}_k - \gamma\boldsymbol{\phi}_{k+1})^\top - \boldsymbol{\phi}_k\mathbb{E}_\mu[F_k\rho_k(\boldsymbol{\phi}_k - \gamma\boldsymbol{\phi}_{k+1})]^\top)}_{\mathbf{A}_{\text{VMETD},k}}\boldsymbol{\theta}_k\}.
\end{aligned}$$

Table 1: Minimum eigenvalues of various algorithms in the 2-state counterexample.

| ALGORITHM | OFF-POLICY TD | TDC | ETD | VMTDC | VMETD |
|---|---|---|---|---|---|
| MINIMUM EIGENVALUES | $-0.2$ | 0.016 | 3.4 | 0.025 | 1.15 |

Therefore,

$$\begin{aligned}
\mathbf{A}_{\text{VMETD}} &= \lim_{k\to\infty}\mathbb{E}[\mathbf{A}_{\text{VMETD},k}] \\
&= \lim_{k\to\infty}\mathbb{E}_\mu[F_k\rho_k\boldsymbol{\phi}_k(\boldsymbol{\phi}_k-\gamma\boldsymbol{\phi}_{k+1})^\top] \\
&\quad -\lim_{k\to\infty}\mathbb{E}_\mu[\boldsymbol{\phi}_k]\mathbb{E}_\mu[F_k\rho_k(\boldsymbol{\phi}_k-\gamma\boldsymbol{\phi}_{k+1})]^\top \\
&= \lim_{k\to\infty}\mathbb{E}_\mu[\boldsymbol{\phi}_k F_k\rho_k(\boldsymbol{\phi}_k-\gamma\boldsymbol{\phi}_{k+1})^\top] \\
&\quad -\lim_{k\to\infty}\mathbb{E}_\mu[\boldsymbol{\phi}_k]\mathbb{E}_\mu[F_k\rho_k(\boldsymbol{\phi}_k-\gamma\boldsymbol{\phi}_{k+1})]^\top \\
&= \lim_{k\to\infty}\mathbb{E}_\mu[\boldsymbol{\phi}_k F_k\rho_k(\boldsymbol{\phi}_k-\gamma\boldsymbol{\phi}_{k+1})^\top] \\
&\quad -\lim_{k\to\infty}\mathbb{E}_\mu[\boldsymbol{\phi}_k]\lim_{k\to\infty}\mathbb{E}_\mu[F_k\rho_k(\boldsymbol{\phi}_k-\gamma\boldsymbol{\phi}_{k+1})]^\top \\
&= \textstyle\sum_s d_\mu(s)\lim_{k\to\infty}\mathbb{E}_\mu[F_k|S_k=s]\mathbb{E}_\mu[\rho_k\boldsymbol{\phi}_k(\boldsymbol{\phi}_k-\gamma\boldsymbol{\phi}_{k+1})^\top|S_k=s] \\
&\quad -\textstyle\sum_s d_\mu(s)\boldsymbol{\phi}(s)\sum_s d_\mu(s)\lim_{k\to\infty}\mathbb{E}_\mu[F_k|S_k=s] \\
&\qquad \mathbb{E}_\mu[\rho_k(\boldsymbol{\phi}_k-\gamma\boldsymbol{\phi}_{k+1})^\top|S_k=s] \\
&= \textstyle\sum_s f(s)\mathbb{E}_\pi[\boldsymbol{\phi}_k(\boldsymbol{\phi}_k-\gamma\boldsymbol{\phi}_{k+1})^\top|S_k=s] \\
&\quad -\textstyle\sum_s d_\mu(s)\boldsymbol{\phi}(s)\sum_s f(s)\mathbb{E}_\pi[(\boldsymbol{\phi}_k-\gamma\boldsymbol{\phi}_{k+1})^\top|S_k=s] \\
&= \textstyle\sum_s f(s)\boldsymbol{\phi}(s)(\boldsymbol{\phi}(s)-\gamma\sum_{s'}[\mathbf{P}_\pi]_{ss'}\boldsymbol{\phi}(s'))^\top \\
&\quad -\textstyle\sum_s d_\mu(s)\boldsymbol{\phi}(s)*\sum_s f(s)(\boldsymbol{\phi}(s)-\gamma\sum_{s'}[\mathbf{P}_\pi]_{ss'}\boldsymbol{\phi}(s'))^\top \\
&= \boldsymbol{\Phi}^\top\mathbf{F}(\mathbf{I}-\gamma\mathbf{P}_\pi)\boldsymbol{\Phi}-\boldsymbol{\Phi}^\top\mathbf{d}_\mu\mathbf{f}^\top(\mathbf{I}-\gamma\mathbf{P}_\mu)\boldsymbol{\Phi} \\
&= \boldsymbol{\Phi}^\top(\mathbf{F}-\mathbf{d}_\mu\mathbf{f}^\top)(\mathbf{I}-\gamma\mathbf{P}_\pi)\boldsymbol{\Phi} \\
&= \boldsymbol{\Phi}^\top(\mathbf{F}(\mathbf{I}-\gamma\mathbf{P}_\pi)-\mathbf{d}_\mu\mathbf{f}^\top(\mathbf{I}-\gamma\mathbf{P}_\pi))\boldsymbol{\Phi} \\
&= \boldsymbol{\Phi}^\top(\mathbf{F}(\mathbf{I}-\gamma\mathbf{P}_\pi)-\mathbf{d}_\mu\mathbf{d}_\mu^\top)\boldsymbol{\Phi},
\end{aligned}$$

$$\begin{aligned}
\mathbf{b}_{\text{VMETD}} &= \lim_{k\to\infty}\mathbb{E}[\mathbf{b}_{\text{VMETD},k}] \\
&= \lim_{k\to\infty}\mathbb{E}_\mu[F_k\rho_k R_{k+1}\boldsymbol{\phi}_k] \\
&\quad -\lim_{k\to\infty}\mathbb{E}_\mu[\boldsymbol{\phi}_k]\mathbb{E}_\mu[F_k\rho_k R_{k+1}] \\
&= \lim_{k\to\infty}\mathbb{E}_\mu[\boldsymbol{\phi}_k F_k\rho_k R_{k+1}] \\
&\quad -\lim_{k\to\infty}\mathbb{E}_\mu[\boldsymbol{\phi}_k]\mathbb{E}_\mu[\boldsymbol{\phi}_k]\mathbb{E}_\mu[F_k\rho_k R_{k+1}] \\
&= \lim_{k\to\infty}\mathbb{E}_\mu[\boldsymbol{\phi}_k F_k\rho_k R_{k+1}] \\
&\quad -\lim_{k\to\infty}\mathbb{E}_\mu[\boldsymbol{\phi}_k]\lim_{k\to\infty}\mathbb{E}_\mu[F_k\rho_k R_{k+1}] \\
&= \textstyle\sum_s f(s)\boldsymbol{\phi}(s)r_\pi-\sum_s d_\mu(s)\boldsymbol{\phi}(s)*\sum_s f(s)r_\pi \\
&= \boldsymbol{\Phi}^\top(\mathbf{F}-\mathbf{d}_\mu\mathbf{f}^\top)\mathbf{r}_\pi.
\end{aligned}$$

Therefore, in the 2-state counterexample, $\mathbf{A}_{\text{VMETD}} = 1.15$, meaning that VMETD can stably converge and converges slower than ETD. However, subsequent experiments showed that the VMETD algorithm converges more smoothly and performs better in controlled experiments.

## Theoretical Analysis

The purpose of this section is to establish the stabilities of the VMTDC algorithm and the VMETD algorithm.

**Theorem 1.** *(Convergence of VMTDC). In the case of off-policy learning, consider the iterations (7), (6) and (5) of VMTDC. Let the step-size sequences $\alpha_k$, $\zeta_k$ and $\beta_k$, $k \geq 0$ 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 $\alpha_k = o(\zeta_k)$, $\zeta_k = o(\beta_k)$. Assume that $(\boldsymbol{\phi}_k, r_k, \boldsymbol{\phi}'_k)$ is an i.i.d. sequence with uniformly bounded second moments. Let $\boldsymbol{A} = \mathrm{Cov}(\boldsymbol{\phi}, \boldsymbol{\phi}-\gamma\boldsymbol{\phi}')$, $\boldsymbol{b} = \mathrm{Cov}(r, \boldsymbol{\phi})$, and $\boldsymbol{C} = \mathbb{E}[\boldsymbol{\phi}\boldsymbol{\phi}^\top]$. Assume that $\boldsymbol{A}$ and $\boldsymbol{C}$ are non-singular matrices. Then the parameter vector $\boldsymbol{\theta}_k$ converges with probability one to $\boldsymbol{A}^{-1}\boldsymbol{b}$.*

*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.

First, note that recursion (5) and (6) 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 (5) 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 (7) are uniformly larger than those in (6) and the increments in iteration (6) are uniformly larger than those in (5), thus (7) is the fastest recursion, (6) is the second fast recursion and (5) is the slower recursion. Along the fastest time scale, iterations of (5), (6) and (7) are associated to ODEs system as follows:

$$\dot{\theta}(t) = 0, \tag{14}$$

$$\dot{u}(t) = 0, \tag{15}$$

$$\dot{\omega}(t) = \mathbb{E}[\delta_t|u(t),\theta(t)] - \omega(t). \tag{16}$$

Based on the ODE (14) and (15), 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 $\theta$ that depends on the initial condition $\theta_0$ of recursion (5) 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 (6). Thus, the ODE pair (14)-(refomegavmtdcFastest) can be written as

$$\dot{\omega}(t) = \mathbb{E}[\delta_t|u,\theta] - \omega(t). \tag{17}$$

Consider the function $h(\omega) = \mathbb{E}[\delta|\theta,u] - \omega$, i.e., the driving vector field of the ODE (17). 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 (17), $\omega^* = \mathbb{E}[\delta|\theta,u]$ is the unique globally asymptotically stable equilibrium. For the ODE

$$\dot{\omega}(t) = h_\infty(\omega(t)) = -\omega(t), \tag{18}$$

apply $\vec{V}(\omega) = (-\omega)^\top(-\omega)/2$ as its associated strict Liapunov function. Then, the origin of (18) is a globally asymptotically stable equilibrium.

Consider now the recursion (7). 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 $\phi_k$, $r_k$, and $\phi'_k$ 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 and Meyn 2000) are verified. Furthermore, Assumptions (TS) of (Borkar and Meyn 2000) is satisfied by our conditions on the step-size sequences $\alpha_k$,$\zeta_k$, $\beta_k$. Thus, by Theorem 2.2 of (Borkar and Meyn 2000) we obtain that $||\omega_k - \omega^*|| \to 0$ almost surely as $k \to \infty$.

Recursion (6) is considered the second timescale. Recursion (5) is considered the slower timescale. For the convergence properties of $u$ and $\theta$, please refer to the appendix. □

**Theorem 2.** *(Convergence of VMETD). In the case of off-policy learning, consider the iterations (2), (13) and (12) of VMETD. Let the step-size sequences $\alpha_k$ and $\beta_k$, $k \geq 0$ 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 $(\boldsymbol{\phi}_k, r_k, \boldsymbol{\phi}'_k)$ is an i.i.d. sequence with uniformly bounded second moments, where $\boldsymbol{\phi}_k$ and $\boldsymbol{\phi}'_k$ are sampled from the same Markov chain. Let $\boldsymbol{A_{VMETD}} = \boldsymbol{\Phi}^\top(\boldsymbol{F}(\boldsymbol{I} - \gamma\boldsymbol{P}_\pi) - \boldsymbol{d}_\mu\boldsymbol{d}_\mu^\top)\boldsymbol{\Phi}$, $\boldsymbol{b_{VMETD}} = \boldsymbol{\Phi}^\top(\boldsymbol{F} - \boldsymbol{d}_\mu\boldsymbol{f}^\top)\boldsymbol{r}_\pi$. Assume that matrix $A$ is non-singular. Then the parameter vector $\boldsymbol{\theta}_k$ converges with probability one to $\boldsymbol{A}_{\boldsymbol{VMETD}}^{-1}\boldsymbol{b_{VMETD}}$.*

*Proof.* The proof of VMETD's convergence is also based on Borkar's Theorem for general stochastic approximation recursions with two time scales (Borkar 1997).

Recursion (13) is considered the faster timescale. For the convergence properties of $\omega$, please refer to the appendix. Recursion (12) is considered the slower timescale. If the key matrix $\mathbf{A}_{\text{VMETD}}$ is positive definite, then $\theta$ converges.

$$\begin{aligned}
(\mathbf{F}(\mathbf{I} - \gamma\mathbf{P}_\pi) - \mathbf{d}_\mu\mathbf{d}_\mu^\top)\mathbf{1} &= \mathbf{F}(\mathbf{I} - \gamma\mathbf{P}_\pi)\mathbf{1} - \mathbf{d}_\mu\mathbf{d}_\mu^\top\mathbf{1} \\
&= \mathbf{F}(\mathbf{1} - \gamma\mathbf{P}_\pi\mathbf{1}) - \mathbf{d}_\mu\mathbf{d}_\mu^\top\mathbf{1} \\
&= (1-\gamma)\mathbf{F}\mathbf{1} - \mathbf{d}_\mu\mathbf{d}_\mu^\top\mathbf{1} \\
&= (1-\gamma)\mathbf{f} - \mathbf{d}_\mu\mathbf{d}_\mu^\top\mathbf{1} \\
&= (1-\gamma)\mathbf{f} - \mathbf{d}_\mu \\
&= (1-\gamma)(\mathbf{I} - \gamma\mathbf{P}_\pi^\top)^{-1}\mathbf{d}_\mu - \mathbf{d}_\mu \\
&= (1-\gamma)[(\mathbf{I} - \gamma\mathbf{P}_\pi^\top)^{-1} - \mathbf{I}]\mathbf{d}_\mu \\
&= (1-\gamma)[\sum_{t=0}^{\infty}(\gamma\mathbf{P}_\pi^\top)^t - \mathbf{I}]\mathbf{d}_\mu \\
&= (1-\gamma)[\sum_{t=1}^{\infty}(\gamma\mathbf{P}_\pi^\top)^t]\mathbf{d}_\mu > 0
\end{aligned} \tag{19}$$

Table 2: Comparison of action selection with and without constant bias in $Q$ values.

| 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 |
| $\arg\min_a Q(s, a)$ | $a_3$ | $a_3$ |

$$\begin{aligned}
\mathbf{1}^\top(\mathbf{F}(\mathbf{I} - \gamma\mathbf{P}_\pi) - \mathbf{d}_\mu\mathbf{d}_\mu^\top) &= \mathbf{1}^\top\mathbf{F}(\mathbf{I} - \gamma\mathbf{P}_\pi) - \mathbf{1}^\top\mathbf{d}_\mu\mathbf{d}_\mu^\top \\
&= \mathbf{d}_\mu^\top - \mathbf{1}^\top\mathbf{d}_\mu\mathbf{d}_\mu^\top \\
&= \mathbf{d}_\mu^\top - \mathbf{d}_\mu^\top \\
&= 0
\end{aligned} \tag{20}$$

(19) and (20) show that the matrix $\mathbf{F}(\mathbf{I} - \gamma\mathbf{P}_\pi) - \mathbf{d}_\mu\mathbf{d}_\mu^\top$ of diagonal entries are positive and its off-diagonal entries are negative. So its each row sum plus the corresponding column sum is positive. So $\mathbf{A}_{\text{VMETD}}$ is positive definite.

□

## Optimal Policy Invariance

This section prove the optimal policy invariance of VMTDC and VMETD in control experiments, laying the groundwork for subsequent experiments.

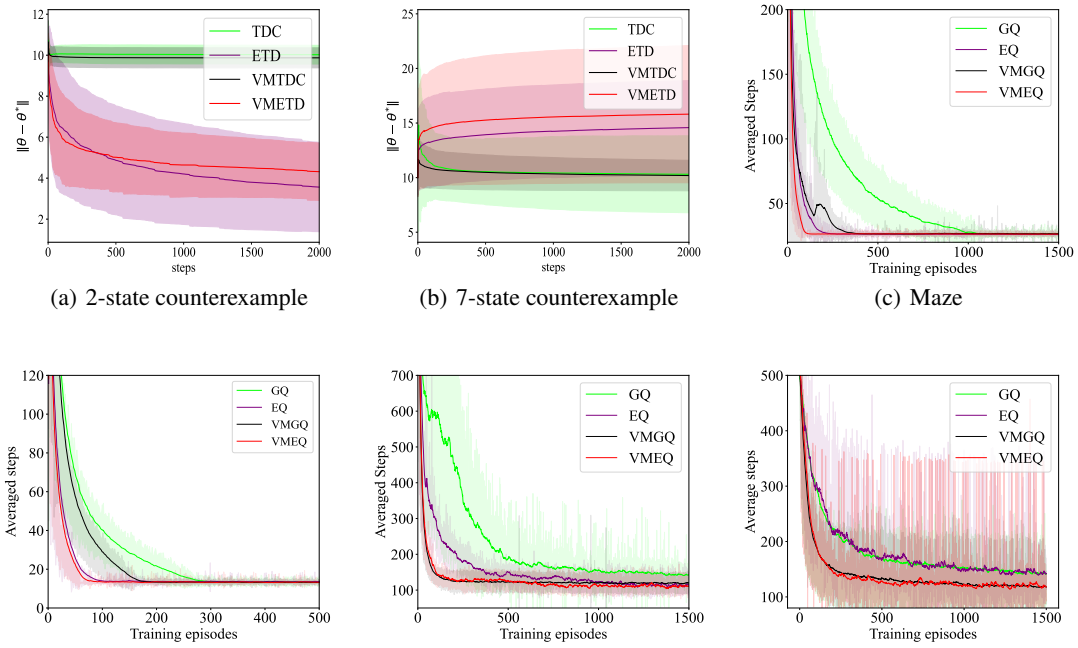
As shown in Table 2, although there is a bias between the true value and the predicted value, action $a_3$ is still chosen under the greedy-policy. On the contrary, supervised learning is usually used to predict temperature, humidity, morbidity, etc. If the bias is too large, the consequences could be 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 (Static PBRS) maintains the policy invariance if the shaping reward follows from $F(s, s') = \gamma f(s') - f(s)$ (Ng, Harada, and Russell 1999).

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

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

where $\omega$ 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 $\omega$ requires expert knowledge. This forces us to learn $\omega$ 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 and Kudenko 2012). However, this result requires




(a) 2-state counterexample (b) 7-state counterexample (c) Maze

(d) Cliff Walking (e) Mountain Car (f) Acrobot

Figure 1: Learning curses of two evaluation environments and four contral environments.

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 $\omega$ 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 $\omega \doteq \mathbb{E}[\delta]$ or $\omega \doteq \mathbb{E}[F\rho\delta]$.

## Experimental Studies

This section assesses algorithm performance through experiments, which are divided into policy evaluation experiments and control experiments. The control algorithms for TDC, ETD, VMTDC, and VMETD are named GQ, EQ, VMGQ, and VMEQ, respectively. The evaluation experimental environments are the 2-state and 7-state counterexample. The control experimental environments are Maze, CliffWalking-v0, MountainCar-v0, and Acrobot-v1. For specific experimental parameters, please refer to the appendix.

For the evaluation experiment, the experimental results align with our previous analysis. In the 2-state counterexample environment, the TDC algorithm has the smallest minimum eigenvalue of the key matrix, resulting in the slowest convergence speed. In contrast, the minimum eigenvalue of VMTDC is larger, leading to faster convergence. Although VMETD's minimum eigenvalue is larger than ETD's, causing VMETD to converge more slowly than ETD in the 2-state counterexample, the standard deviation (shaded area) of VMETD is smaller than that of ETD, indicating that VMETD converges more smoothly. In the 7-state counterexample environment, VMTDC converges faster than TDC and both VMETD and ETD are diverge.

For the control experiments, the results for the maze and cliff walking environments are similar: VMGQ outperforms GQ, EQ outperforms VMGQ, and VMEQ performs the best. In the mountain car and Acrobot experiments, VMGQ and VMEQ show comparable performance, both outperforming GQ and EQ. In summary, for control experiments, VM algorithms outperform non-VM algorithms.

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.

## Conclusion and Future Work

Value-based reinforcement learning typically aims to minimize error as an optimization objective. As an alternation, this study proposes new objective functions: VBE and VPBE, and derives many variance minimization algorithms, including VMTD, VMTDC and VMETD. All 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 and VMETD. (2) extensions of VBE and VPBE to multi-step returns. (3) extensions to nonlinear approximations, such as neural networks.

## References


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

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

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

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

Chen, X.; Ma, X.; Li, Y.; Yang, G.; Yang, S.; and Gao, Y. 2023. Modified Retrace for Off-Policy Temporal Difference Learning. In *Uncertainty in Artificial Intelligence*, 303–312. PMLR.

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

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

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

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

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

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

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

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

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

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

Liu, B.; Liu, J.; Ghavamzadeh, M.; Mahadevan, S.; and Petrik, M. 2016. Proximal Gradient Temporal Difference Learning Algorithms. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 4195–4199.

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

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

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

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

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

Sutton, R. S.; Maei, H. R.; and Szepesvári, C. 2008. A Convergent $O(n)$ Temporal-difference Algorithm for Off-policy Learning with Linear Function Approximation. In *Advances in Neural Information Processing Systems*, 1609–1616. Cambridge, MA: MIT Press.

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

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

Xu, T.; Wang, Z.; Zhou, Y.; and Liang, Y. 2019. Reanalysis of Variance Reduced Temporal Difference Learning. In *International Conference on Learning Representations*.

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