On the Variance of Temporal Difference Learning and its Reduction Using Control Variates

Hsiao-Ru Pan
MPI for Intelligent Systems, Tübingen

Bernhard Schölkopf
MPI for Intelligent Systems, Tübingen

Abstract

We analyze the finite-sample variance of temporal difference (TD) learning in the phased TD setting, and show that one of the mechanisms behind bootstrapping's ability to reduce variance is by effectively aggregating over a larger number of independent trajectories. Based on this insight, we demonstrate that asymptotically, the variance of TD learning is bounded from above by Monte-Carlo (MC) estimators. In addition, we draw connections to Direct Advantage Estimation (DAE), a method for estimating the advantage function, and show that it can be seen as a type of regression-adjusted control variate, which further reduces the variance of TD. Finally, we illustrate the asymptotic behaviors of these estimators empirically with carefully designed environments.

1 Introduction

Policy evaluation, that is, estimating returns from given states (or state-action pairs) is a central problem in reinforcement learning (RL) [Sutton et al., 1998]. Among various estimation methods, temporal difference (TD) learning [Sutton, 1988] stands out as a cornerstone technique for this class of problems. Traditional methods like Monte-Carlo (MC) methods estimate returns by averaging returns from sample trajectories, typically resulting in unbiased but high variance estimates. In contrast, TD learning updates value estimates iteratively through bootstrapping (i.e., estimate based on previous estimates), thereby avoiding the need for full trajectories and tends to exhibit lower variance.Previously, it was shown that bootstrapping can be seen as a form of bias-variance trade-off [Kearns and Singh, 2000], where the high variance estimates from sample trajectories are replaced with low variance bootstrapped values. While this intuition largely holds true, it is not difficult to see that the full story is more complex than this. For example, suppose we initialize the value estimates with the *ground truth* value function and update them using TD learning. If the step-size is non-zero and the rewards are not deterministic, then we essentially *inject* variance into the estimates and increase the estimation error. This shows that the way bootstrapping reduces variance may be more nuanced than simply "replace a high variance trajectory with a biased estimate".

In the present work, we analyze the variance of multi-step TD learning in the phased setting [Kearns and Singh, 1998], which abstracts away some of the complexities due to stochastic approximations, and reveal one of the mechanisms behind bootstrapping's variance reduction property. Beyond TD learning, we also draw connections to Direct Advantage Estimation (DAE) [Pan et al., 2022], a recently proposed method that simultaneously estimates the value function and the advantage function, and show that DAE can be seen as a type of control variate regression that further reduces the variance of TD learning. To summarize, we show that

- Asymptotically, the variance of multi-step TD learning is bounded above by the variance of MC methods.
- The advantage function can be seen as control variates for value estimation, which reduces the variance of multi-step learning, and DAE is a type of regression-adjusted control variate.

18th European Workshop on Reinforcement Learning (EWRL 2025).

Finally, we construct examples to illustrate the asymptotic behaviors of these estimators.

2 Background

We consider a discounted Markov Decision Process [Puterman, 2014] $(\mathcal{S}, \mathcal{A}, p, r, \gamma)$ with finite state space $\mathcal{S}$, finite action space $\mathcal{A}$, transition probability $p(s'|s,a)$, reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and discount factor $\gamma \in [0,1)$. For simplicity, we assume the reward function is deterministic. A policy $\pi(\cdot|s)$ is a function that maps states to distributions over $\mathcal{A}$. The state and state-action value functions are defined by $V^\pi(s) = \mathbb{E}_\pi[\sum_{t=0}^\infty \gamma^t r(s_t,a_t)|s_0{=}s]$, and $Q^\pi(s,a) = \mathbb{E}_\pi[\sum_{t=0}^\infty \gamma^t r(s_t,a_t)|s_0{=}s, a_0{=}a]$, respectively ($\mathbb{E}_\pi$ indicates that actions are sampled from π). The advantage function is defined by $A^\pi(s) = Q^\pi(s,a) - V^\pi(s)$. For the present work, we will consider π as fixed and omit it in the discussion when the context is clear.

The goal of reinforcement learning (RL) [Sutton et al., 1998] is to find the optimal policy through interactions in a given MDP, and the value function is at the core of various policy optimization algorithms. However, the value function is typically unknown a priori, and one central problem in RL is how to learn value functions efficiently. One classical method is TD learning [Sutton, 1988], which updates the value function through bootstrapping. More precisely, TD(0) estimates the value function by sampling (s,a,r,s') tuples and updating the values by $V(s) \leftarrow V(s) + \alpha(r + \gamma V(s') - V(s))$, where $\alpha \in \mathbb{R}$ is the learning rate. One downside of TD(0) is that it only updates the value one-step at a time, which can be inefficient when rewards are delayed. As such, it is common to consider multi-step TD learning by sampling multiple timesteps before updating the value, that is,

$$V(s_0) \leftarrow V(s_0) + \alpha(r_0 + \gamma r_1 + \cdots + \gamma^k V(s_k) - V(s_0)). \tag{1}$$

Phased TD We presently consider the *phased* setting [Kearns and Singh, 1998], which removes some of the complexities of TD learning due to asynchronous updates and stochastic approximations. In phased TD(k) (k denotes the backup horizon), value estimates are updated in *phases*, where each phase consists of (1) sampling n k-step trajectories for each state: $\mathcal{D} = \{\tau_{s,i}\}_{s\in\mathcal{S}, i\in[1,\cdots,n]}$, where $\tau_{s,i} = (s, a_0^i, r_0^i, s_1^i, \cdots, s_k^i)$ (n is assumed to be fixed throughout the paper unless otherwise stated), and (2) updating the value of each state synchronously by

$$V_{\mathrm{TD}(k)}^{T+1}(s) \leftarrow \frac{1}{n}\sum_{i=1}^n \left(r_0^i + \gamma r_1^i + \cdots + \gamma^{k-1} r_{k-1}^i + \gamma^k V_{\mathrm{TD}(k)}^T\left(s_k^i\right)\right) \ \forall s \in \mathcal{S}, \tag{2}$$

where T denotes the phase. It was shown that phased TD is analogous to TD learning with a fixed learning rate under mild assumptions. Kearns and Singh [2000] used this setting to analyze the bias-variance tradeoff of multi-step TD learning, and showed that the estimation error $\Delta_T = \max_s |V^T(s) - V^\pi(s)|$ satisfies

$$\Delta_{T+1} \le \underbrace{\left|\frac{1}{n}\sum_{i=1}^n \left(\sum_{t=0}^{k-1}\gamma^t r_t^i\right) - \mathbb{E}\left[\left(\sum_{t=0}^{k-1}\gamma^t r_t\right)\middle| s_0{=}s\right]\right|}_{\text{variance}} + \gamma^k \underbrace{\Delta_T}_{\text{bias}}.$$

The variance term accounts for the stochasticity from the sample rewards, while the bias term accounts for the error from bootstrapping. The error can be further bounded by a PAC style bound (without loss of generality, assume $\Delta_0 = 1$),

$$\Delta_T \le \frac{1-\gamma^{kT}}{1-\gamma}\sqrt{\frac{3\log(k/\delta)}{n}} + \gamma^{kT}, \quad \lim_{T\to\infty}\Delta_T \le \frac{1}{1-\gamma}\sqrt{\frac{3\log(k/\delta)}{n}} \text{ (asymptotic)}, \tag{3}$$

with probability $1-\delta$. Intuitively, increasing k reduces the bias from bootstrapping at the cost of increased variances from the rewards. However, this bias-variance decomposition fails to consider the variance from finite samples of s_k^i (merged into the bias term) and the variance from the previous estimate $V_{\mathrm{TD}(k)}^T$. Furthermore, as $k \to \infty$ (MC estimation), this bound becomes vacuous even though the rewards are bounded.

Control Variate Control variate [Asmussen and Glynn, 2007] is a technique for reducing variance of MC simulation. We briefly review the basics of control variate in the one-dimensional setting, and refer the reader to Owen [2013] for a more general treatment.

Recall that we can estimate $\mu_X = \mathbb{E}[X]$ using MC methods by $\hat{\mu}_X = \frac{1}{n}\sum_{i=1}^n X_i$, where $X_i \overset{\text{iid}}{\sim} X$. We now introduce another random variable Y, called the control variate, that is correlated with X and has known $\mathbb{E}[Y]$ (assume $\mathbb{E}[Y] = 0$, otherwise use $Z = Y - \mathbb{E}[Y]$), and define a new estimator

$$\hat{\mu}_{X,\lambda Y} = \frac{1}{n}\sum_{i=1}^n X_i + \lambda Y_i, \tag{4}$$

where $(X_i, Y_i) \overset{\text{iid}}{\sim} (X, Y)$ and λ is a tunable constant. This new estimator remains unbiased, but has a different variance $\text{Var}(\hat{\mu}_{X,\lambda Y}) = \text{Var}(\hat{\mu}_X) + \frac{\lambda^2}{n}\text{Var}(Y) + \frac{2\lambda}{n}\text{Cov}(X, Y)$. If we choose $\lambda = \lambda^* = -\frac{\text{Cov}(X,Y)}{\text{Var}(Y)}$, then the variance becomes $\text{Var}(\hat{\mu}_{X,\lambda^* Y}) = \text{Var}(\hat{\mu}_X) - \frac{\text{Cov}(X,Y)^2}{n\text{Var}(Y)} \leq \text{Var}(\hat{\mu}_X)$, which is never worse than the original estimator. Intuitively, one can view $\mathbb{E}[Y]$ as our prior knowledge of the distribution of X, and control variate provides us a way to exploit this knowledge to reduce variance.In practice, however, λ^* is usually unknown, and has to be estimated from data. One simple choice is to replace the variances by their estimators, that is, $\hat{\lambda} = -\frac{s_{X,Y}}{s_Y}$. Interestingly, one can show that this is equivalent to solving the following least squares problem,

$$\sum_{i=1}^n (\theta - x_i - \lambda y_i)^2, \tag{5}$$

where θ is the intercept of the linear model. It turns out that the minimizer is $(\theta, \lambda) = (\hat{\mu}_{X,\hat{\lambda}Y}, \hat{\lambda})$. This approach is also known as regression-adjusted control variate, and can be readily generalized to more complex settings (e.g., multiple control variates). The error of this estimator is:

$$\hat{\mu}_{X,\hat{\lambda}Y} - \mu_X = \underbrace{(\hat{\lambda} - \lambda^*)\sum_{i=1}^n \frac{Y_i}{n}}_{O(1/n)} + \underbrace{\hat{\mu}_{X,\lambda^* Y} - \mu_X}_{O(1/\sqrt{n})}. \tag{6}$$

As both $\hat{\lambda} - \lambda^*$ and $\sum_{i=1}^n \frac{Y_i}{n}$ approach zero with rates $O(1/\sqrt{n})$, their product converges to 0 with rate $O(1/n)$. Consequently, the second term dominates the error asymptotically, and $\hat{\mu}_{X,\lambda^* Y}$ can be seen as a first-order approximation of $\hat{\mu}_{X,\hat{\lambda}Y}$. Finally, we note that the estimator is, in general, biased since Y_i can be correlated with $\hat{\lambda}$ (i.e., $\mathbb{E}[Y_i\hat{\lambda}] \neq 0$); however, the estimator remains consistent as the errors approach zero by the law of large numbers.

Direct Advantage Estimation Direct Advantage Estimation (DAE) [Pan et al., 2022] is a method developed for estimating the advantage function directly from sampled trajectories. Similar to multi-step TD, DAE can update values by bootstrapping previous estimates. More specifically, DAE estimates the values by iteratively minimizing the following constrained least-squares

$$\mathcal{L}(\hat{A}, \hat{V}) = \mathbb{E}_\pi\left[\left(\sum_{t=0}^{k-1}\gamma^t(r_t - \hat{A}_t) + \gamma^k V_{\text{target}}(s_k) - \hat{V}(s_0)\right)^2\right] \tag{7}$$

$$(A^*, V^*) = \underset{\hat{A}\in F_\pi, \hat{V}}{\arg\min}\, \mathcal{L}_T(\hat{A}, \hat{V}), \quad F_\pi = \{f | \mathbb{E}_\pi\left[f(s,a)|s\right] = 0\ \forall s\}. \tag{8}$$

It was shown that iteratively updating the target and minimizing this objective converges to (A^π, V^π). In practice, the expectation is replaced with an average over sample trajectories. We note that, if we force $\hat{A} \equiv 0$ and only optimize with respect to $\hat{V}$, then DAE reduces to multi-step TD. DAE demonstrated strong empirical performance in the deep RL setting; however, it remains unclear whether estimating the value function this way is beneficial compared to classical approaches.

3 Variance of Monte Carlo Methods

In this section, we show that the variance of the (first-visit) MC estimator can be broken down into segments, which will become useful later when comparing to multi-step TD.

Table 1: Variables names and their definitions. Note that $\bar{\mathbf{P}}$ is a (row) vector of size $|\mathcal{S}|$.

Variable	Def.	Description
$\mathbf{G}$	$\frac{1}{n}\sum_{i=1}^{n}\sum_{t=0}^{\infty}\gamma^t r_t^i$	Average of sample returns
$\mathcal{D}$	$\left\{(s, a_0^i, r_0^i, s_1^i, \cdots, s_k^i)\right\}$	k-step partial trajectories
$\bar{\mathbf{R}}$	$\frac{1}{n}\sum_{i=1}^{n}\sum_{t=0}^{k-1}\gamma^t r_t^i$	Average of sample k-step returns
$\bar{\mathbf{P}}$	$\bar{\mathbf{P}}_{s'} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(s_k^i = s')$	Empirical k-step transition distribution

Recall that MC methods estimate values by averaging returns from sample trajectories:

$$V_{\mathrm{MC}}(s) = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{t=0}^{\infty}\gamma^t r_t^i\right), \tag{9}$$

where the superscript i denote the ith trajectory. In Table 1, we introduce simplified notations to ease the presentation. It should be noted that the variables are random variables (vectors) based on the sampled trajectories at each phase.

We first state a lemma which shows that the variance of the MC estimator can be decomposed into the sum of the variances of partial trajectories.

Lemma 1. $\mathrm{Var}(V_{\mathrm{MC}}(s)) = \sum_{m=0}^{\infty}\gamma^{2km}\mathbb{E}\left[\mathrm{Var}(\bar{\mathbf{R}} + \gamma^k\bar{\mathbf{P}}\mathbf{V}^\pi|s_{km})\middle|\, s_0{=}s\right]$, *where* $\mathbf{V}^\pi \in \mathbb{R}^{|\mathcal{S}|}$ *is the true value function.*

Proof. Note that, when conditioned on $\mathcal{D}$, both $\mathbf{R}$ and $\mathbf{P}$ become constants. Using this fact, we have the following recurrence relation:

$$\begin{aligned}
&\mathrm{Var}(V_{\mathrm{MC}}(s)) = \mathrm{Var}\left(\bar{\mathbf{G}}\middle|\, s_0{=}s\right)\\
&= \mathrm{Var}\left(\mathbb{E}\left[\bar{\mathbf{G}}\middle|\,\mathcal{D}, s_0{=}s\right]\middle|\, s_0{=}s\right) + \mathbb{E}\left[\mathrm{Var}\left(\bar{\mathbf{G}}\middle|\,\mathcal{D}, s_0{=}s\right)\middle|\, s_0{=}s\right]\\
&= \mathrm{Var}\left(\bar{\mathbf{R}} + \gamma^k\bar{\mathbf{P}}\mathbf{V}^\pi\middle|\, s_0{=}s\right) + \mathbb{E}\left[\mathrm{Var}\left(\frac{1}{n}\sum_{i=1}^{n}\sum_{t=k}^{\infty}\gamma^t r_t^i\middle|\,\mathcal{D}\right)\middle|\, s_0{=}s\right]\\
&= \mathrm{Var}\left(\bar{\mathbf{R}} + \gamma^k\bar{\mathbf{P}}\mathbf{V}^\pi\middle|\, s_0{=}s\right) + \frac{1}{n}\mathbb{E}\left[\sum_{i=1}^{n}\mathrm{Var}\left(\gamma^k V_{\mathrm{MC}}(s_k^i)\right)\middle|\, s_0{=}s\right]\\
&= \mathrm{Var}\left(\bar{\mathbf{R}} + \gamma^k\bar{\mathbf{P}}\mathbf{V}^\pi\middle|\, s_0{=}s\right) + \gamma^{2k}\mathbb{E}\left[\mathrm{Var}\left(V_{\mathrm{MC}}(s_k)\right)|s_0{=}s\right].
\end{aligned}$$

Since $\mathrm{Var}(V_{\mathrm{MC}}(\cdot))$ is bounded, Lemma 1 follows from expanding this recursion. □

We abused the notation slightly by denoting the variance starting from state s_{km} as $\mathrm{Var}(\cdot|s_{km})$ to avoid confusion with the s_0 in the outer conditional expectation. As we will see, the variance of TD can be similarly decomposed with $\mathbf{V}^\pi$ replaced by bootstrapping values.

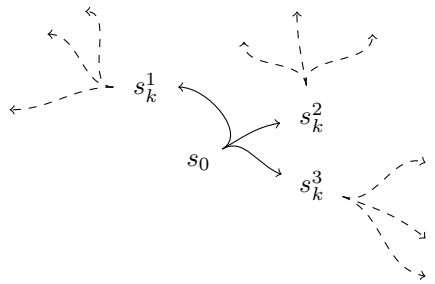
4 Variance of Multi-Step TD Learning

In this section, we show that the variance decomposition presented in Lemma 1 is closely related to the variance of TD learning, and use it to analyze the variance of TD learning.

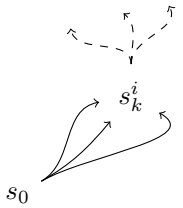
We first rewrite Equation 2 into the following:

$$V_{\mathrm{TD}(k)}^T(s) = \bar{\mathbf{R}}^T + \gamma^k\bar{\mathbf{P}}^T\mathbf{V}_{\mathrm{TD}(k)}^{T-1},$$

where the superscript T indicates the phase of the corresponding variables, and $\mathbf{V}_{\mathrm{TD}(k)}^T \in \mathbb{R}^{|\mathcal{S}|}$ is the random vector with entries $V_{\mathrm{TD}(k)}^T(s)$. Without loss of generality, we set $\mathbf{V}_{\mathrm{TD}(k)}^0 \equiv 0$. Note that $\mathbf{V}_{\mathrm{TD}(k)}^T$ is now a random vector for $T > 0$ as the update involves random variables $\bar{\mathbf{R}}$ and $\bar{\mathbf{P}}$.



(a) Independent bootstrapping states



(b) Correlated bootstrapping states

Figure 1: Illustration of how TD can reduce variance. Solid and dashed arrows represent trajectories collected in the current phase and the previous phase, respectively. (a) The bootstrapping states s_k^i are independent, and averaging over their values effectively averages over 9 independent trajectories collected in the previous phase. (b) The bootstrapping states s_k^i are all the same, and averaging their values provides no variance reduction, as the number of independent trajectories remains 3.

Let us first consider the expectation of $\mathbf{V}_{\mathrm{TD}}^T$ (note that $\mathbf{P}^T$ and $\mathbf{V}_{\mathrm{TD}}^{T-1}$ are independent):

$$\begin{aligned}\bar{V}_{\mathrm{TD}(k)}^T(s) &\coloneqq \mathbb{E}\left[V_{\mathrm{TD}(k)}^T(s)\right] = \mathbb{E}\left[\bar{\mathbf{R}}^T\right] + \gamma^k \mathbb{E}\left[\bar{\mathbf{P}}^T\right] \mathbb{E}\left[\mathbf{V}_{\mathrm{TD}(k)}^{T-1}\right] \\ &= \mathbb{E}\left[\sum_{t=0}^{k-1} \gamma^t r_t + \gamma^k \bar{V}_{\mathrm{TD}(k)}^{T-1}(s_k) \middle| s_0{=}s\right],\end{aligned}$$

and we recover the k-step Bellman update, implying that the bias of TD decays exponentially fast. Next, similar to Lemma 1, we show that the upper bound of the variance of $\mathbf{V}_{\mathrm{TD}}$ also satisfies a recurrence relation.

Lemma 2. $\mathrm{Var}(V_{\mathrm{TD}}^T(s)) \le \mathrm{Var}\left(\bar{\mathbf{R}}^T + \gamma^k \bar{\mathbf{P}}^T \bar{\mathbf{V}}_{\mathrm{TD}}^{T-1} \middle| s_0{=}s\right) + \gamma^{2k} \mathbb{E}\left[\mathrm{Var}\left(V_{\mathrm{TD}}^{T-1}(s_k)\right) \middle| s_0{=}s\right]$.

Proof.

$$\begin{aligned}&\mathrm{Var}(V_{\mathrm{TD}(k)}^T(s)) = \mathrm{Var}\left(\bar{\mathbf{R}}^T + \gamma^k \bar{\mathbf{P}}^T \mathbf{V}_{\mathrm{TD}(k)}^{T-1} \middle| s_0{=}s\right) \\ &= \mathrm{Var}\left(\mathbb{E}\left[\bar{\mathbf{R}}^T + \gamma^k \bar{\mathbf{P}}^T \mathbf{V}_{\mathrm{TD}(k)}^{T-1} \middle| \mathcal{D}\right] \middle| s_0{=}s\right) + \mathbb{E}\left[\mathrm{Var}\left(\bar{\mathbf{R}}^T + \gamma^k \bar{\mathbf{P}}^T \mathbf{V}_{\mathrm{TD}(k)}^{T-1} \middle| \mathcal{D}\right) \middle| s_0{=}s\right] \\ &= \mathrm{Var}\left(\bar{\mathbf{R}}^T + \gamma^k \bar{\mathbf{P}}^T \bar{\mathbf{V}}_{\mathrm{TD}(k)}^{T-1} \middle| s_0{=}s\right) + \gamma^{2k} \mathbb{E}\left[\mathrm{Var}\left(\bar{\mathbf{P}}^T \mathbf{V}_{\mathrm{TD}(k)}^{T-1} \middle| \mathcal{D}\right) \middle| s_0{=}s\right] \\ &\le \mathrm{Var}\left(\bar{\mathbf{R}}^T + \gamma^k \bar{\mathbf{P}}^T \bar{\mathbf{V}}_{\mathrm{TD}(k)}^{T-1} \middle| s_0{=}s\right) + \gamma^{2k} \mathbb{E}\left[\bar{\mathbf{P}}^T \mathrm{Var}\left(\mathbf{V}_{\mathrm{TD}(k)}^{T-1}\right) \middle| s_0{=}s\right] \qquad \text{(Jensen's ineq.)} \\ &= \mathrm{Var}\left(\bar{\mathbf{R}}^T + \gamma^k \bar{\mathbf{P}}^T \bar{\mathbf{V}}_{\mathrm{TD}(k)}^{T-1} \middle| s_0{=}s\right) + \gamma^{2k} \mathbb{E}\left[\mathrm{Var}\left(V_{\mathrm{TD}(k)}^{T-1}(s_k)\right) \middle| s_0{=}s\right].\end{aligned}$$

□

Expanding this recursion gives us:

$$\mathrm{Var}(V_{\mathrm{TD}(k)}^T(s)) \le \sum_{m=0}^{T-1} \gamma^{2km} \mathbb{E}\left[\mathrm{Var}\left(\bar{\mathbf{R}}^{T-m} + \gamma^k \bar{\mathbf{P}}^{T-m} \bar{\mathbf{V}}_{\mathrm{TD}(k)}^{T-1-m} \middle| s_{km}\right) \middle| s_0{=}s\right],$$

and we arrive at a similar expression to the MC estimator (Lemma 1). In fact, we get, in the limit:

Theorem 1. $\lim_{T\to\infty} \mathrm{Var}(V_{\mathrm{TD}(k)}^T(s)) \le \mathrm{Var}(V_{\mathrm{MC}}(s))$.

See Appendix A.1 for a proof. This bound shows that, asymptotically, TD learning is no worse than MC methods, independent of the backup length k; however, it also suggests that TD learning, in the worst case, can suffer from the same variance as MC methods. To understand when this could happen, let us examine the only inequality used in the derivation, namely, $\mathrm{Var}\left(\bar{\mathbf{P}}\mathbf{V}_{\mathrm{TD}(k)}^{T-1} \middle| \mathcal{D}\right) \le \bar{\mathbf{P}}\mathrm{Var}\left(\mathbf{V}_{\mathrm{TD}(k)}^{T-1}\right)$. Note that the left-hand side is simply the variance of the average of the bootstrap

values, and this inequality becomes an equality when the value estimates have correlation 1 (e.g., s_k^i are the same for all i). This suggests that one way bootstrapping reduces variance is by effectively aggregating over a larger pool of independent trajectories, as illustrated in Figure 1. In Section 7, we also construct toy examples to illustrate this effect.

5 Control variate and the advantage function

We now turn to the question of how we can use control variates to reduce variance. We begin by considering the π-centered function class $F_\pi = \{f | \mathbb{E}_\pi [f(s,a)|s] = 0 \; \forall s\}$, which has the following property:

$$\mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) - f(s_t, a_t) \right) \right] = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right] \quad \forall f \in F_\pi.$$

In other words, introducing f does not bias the MC estimate, and f can be seen as a control variate. A natural question is, then, what would be the optimal choice of f^* that minimizes the variance, i.e.,

$$f^* = \arg\min_{f \in F_\pi} \mathrm{Var} \left(\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) - f(s_t, a_t) \right) \right)$$

Pan et al. [2022] proved that the advantage function A^π is the unique minimizer of this variance under mild coverage assumptions on the policy. Now, if the advantage function is known, we can combine it with the MC estimator to reduce its variance via

$$V_{\mathrm{MC-A}}(s) = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{t=0}^{\infty} \gamma^t \left(r_t^i - A^\pi(s_t^i, a_t^i) \right) \right). \tag{10}$$

One may wonder to what extent can this control variate reduce the variance of MC estimates. To answer this, we use the return decomposition proposed by Pan and Schölkopf [2024]

$$\sum_{t=0}^{\infty} \gamma^t r_t = V^\pi(s_0) + \sum_{t=0}^{\infty} \gamma^t (A^\pi(s_t, a_t) + B^\pi(s_t, a_t, r_t, s_{t+1})),$$

where $B^\pi(s_t, a_t, r_t, s_{t+1}) = r_t + \gamma V^\pi(s_{t+1}) - \mathbb{E}[r + \gamma V^\pi(s') | s_t, a_t]$. Now, if $B^\pi \equiv 0$ (e.g., deterministic environment), then the decomposition reduces to

$$\sum_{t=0}^{\infty} \gamma^t r_t = V^\pi(s_0) + \sum_{t=0}^{\infty} \gamma^t A^\pi(s_t, a_t). \tag{11}$$

Combine this with Equation 10, we have $\mathrm{Var}(V_{\mathrm{MC-A}}(s)) = \mathrm{Var}(V^\pi(s)) = 0$. This means that A^π can fully explain the variance, and we suffer no variance by using the MC estimator in this case. For more general environments, B^π is required to account for the variance caused by stochastic transitions, and it remains open whether B^π can be easily estimated in model-free settings. As such, we focus on the advantage function in the present work.

In practice, the advantage function is rarely known a priori, and we have to estimate both the value function and the advantage function simultaneously. In Section 6, we show that Direct Advantage Estimation (DAE) [Pan et al., 2022] can be seen as a type of regression-adjusted control variate, which achieves this.

6 Direct Advantage Estimation and Control Variate Regression

Recall that DAE estimates the value function and the advantage function by solving a constrained least-square problem (Equation 7). Similar to TD learning, DAE can also bootstrap with previous estimates. For the present work, we focus only on the value estimate and treat $\hat{A}$ as nuisance parameters. This allows us to remove the constraint and reformulate the empirical objective into:

$$\frac{1}{n} \sum_{i=1}^{n} \left(\sum_{t=0}^{k-1} \gamma^t \left(r_t^i - \left(\hat{A}_t^i - \sum_a \pi(a|s_t) \hat{A}(s_t^i, a) \right) \right) + \gamma^k V_{\mathrm{DAE}(k)}^T(s_k^i) - \hat{V}(s_0) \right)^2,$$

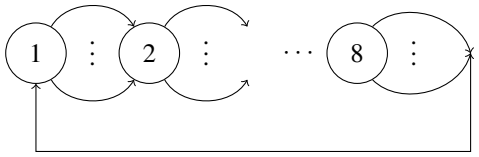


Figure 2: Chain MDP with $\mathcal{S} = \{1, 2, ..., 8\}$, $\mathcal{A} = \{1, ..., |\mathcal{A}|\}$, and $r(a) = \frac{(-1)^a}{4}$ (independent of state). After state 8, the agent returns to state 1.

Figure 3: Parameters of the experiment.

Param.	Description
$\lvert\mathcal{A}\rvert$	action space size
n	number of sample trajectories
k	backup length
p_r	probability of reward masking
p_s	probability of sticky transition

where $\hat{A}_t^i = \hat{A}(s_t^i, a_t^i)$. The update rule then becomes solving for the minimizing $(\hat{A}, \hat{V})$ of this empirical objective. Under this formulation, $A^{T+1}_{\mathrm{DAE}(k)}$ may no longer be unique, but $V^{T+1}_{\mathrm{DAE}(k)}$ remains unchanged. Let us now consider DAE in the phased setting, which turns out to have a similar update rule as TD learning. We first introduce $\mathbf{M} \in \mathbb{R}^{n \times |\mathcal{S}||\mathcal{A}|}$:

$$\mathbf{M}_{i,(s,a)} = \sum_{t=0}^{k-1} \gamma^t \left(\mathbb{I}(s_t^i{=}s, a_t^i{=}a) - \pi(a|s)\mathbb{I}(s_t^i{=}s) \right). \tag{12}$$

Essentially, this matrix compares the empirical occupancy measure of each trajectory to the occupancy measure given the policy. Notice that the finite-sample phased DAE update can be written as:

$$(\mathbf{A}^T, V^T_{\mathrm{DAE}(k)}(s)) = \operatorname*{arg\,min}_{\hat{A}, \hat{V}(s)} \sum_{i=1}^{n} \left(\sum_{t=0}^{k-1} \gamma^t r_t^i + \gamma^k \mathbf{V}^{T-1}_{\mathrm{DAE}(k)}(s_k^i) - \mathbf{M}_i^{T-1} \hat{\mathbf{A}} - \hat{V}(s) \right)^2, \tag{13}$$

where $\hat{\mathbf{A}} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ is a parameter vector. Comparing this to Equation 5, we see that DAE is a case of regression-adjusted control variate, where $\mathbf{M}$ is the control variate with $\mathbb{E}[\mathbf{M}] = 0$ and corresponding coefficients $\hat{\mathbf{A}}$. Let $\bar{\mathbf{M}} \coloneqq \frac{1}{n}\sum_{i=1}^n \mathbf{M}_i$, then the update rule is equal to:

$$V^T_{\mathrm{DAE}(k)}(s) = \bar{\mathbf{R}}^T + \gamma^k \bar{\mathbf{P}}^T \mathbf{V}^{T-1}_{\mathrm{DAE}} - \bar{\mathbf{M}}^T \mathbf{A}^T.$$

As pointed out in Section 2, the control variate regression estimator behaves similarly to the one with optimal control variate coefficients up to first-order approximation. Consequently, we have $V^T_{\mathrm{DAE}(k)}(s) \approx V^{*T}_{\mathrm{DAE}(k)}(s)$, where V^{*T}_{DAE} denotes the estimator with optimal control variate coefficients. One can then show that:

Theorem 2. $\limsup_{T\to\infty} \mathrm{Var}(V^{*T}_{\mathrm{DAE}}(s)) \le \mathrm{Var}(V_{\mathrm{MC-A}}(s)) \le \mathrm{Var}(V_{\mathrm{MC}}(s))$.

See Appendix A.2 for a proof. Comparing this to Corollary 1, we find that, asymptotically and up to first-order approximation, DAE enjoys a lower upper bound on the variance by using control variates.

7 Empirical Illustration

In this section, we illustrate the behaviors of different estimators through experiments based on variants of the chain environment shown in Figure 2. Despite its simplicity, the environment is sufficiently expressive to elucidate various properties pertaining to the estimators analyzed in the present study.

All experiments are based on the phased setting, where values are updated synchronously at the end of each phase. We fix the policy π to be uniform, the discount factor at $\gamma = 0.99$, and consider two types of stochasticity:

1. Reward masking: The rewards are masked out with probability p_r

$$p(r(a)) = \begin{cases} p_r & r(a) = 0 \\ 1 - p_r & r(a) = \frac{(-1)^a}{4} \end{cases}$$

2. Sticky transition: With probability p_s, the agent stays in the current state instead of advancing to the next state

$$p(s_{t+1}|s_t) = \begin{cases} p_s & s_{t+1} = s_t \\ 1 - p_s & s_{t+1} = s_t \bmod 8 + 1 \end{cases}$$

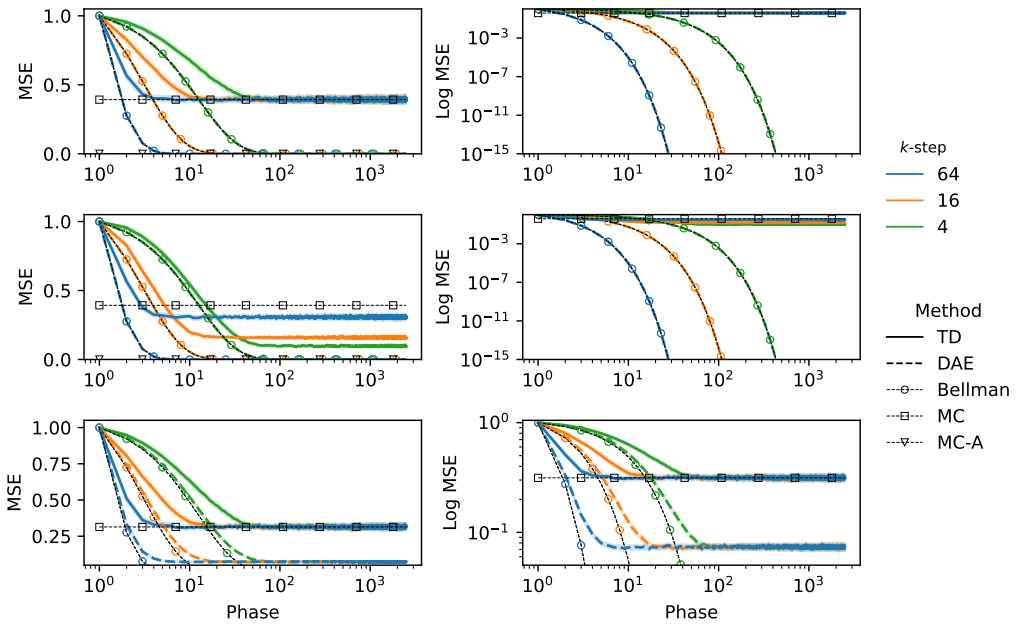


Figure 4: The deterministic case (top), the sticky transition case (middle), and the reward masking case (bottom). Lines and shadings represent (mean $\pm$ 3 standard error).

For simplicity, we consider cases where $|\mathcal{A}|$ is even, such that $V^\pi \equiv 0$, and the variances of the MC and the MC-A estimators are equal to

$$\mathrm{Var}(V_{\mathrm{MC}}(s)) = \frac{(1-p_r)}{16(1-\gamma^2)n}, \quad \text{and} \quad \mathrm{Var}(V_{\mathrm{MC-A}}(s)) = p_r \mathrm{Var}(V_{\mathrm{MC}}(s)). \tag{14}$$

We note that the variances do not depend on p_s due to the symmetric nature of the states.

For the following experiments, the number of phases is fixed at 2500, which we found sufficient for the estimators to converge, and each run (configuration) is repeated for 1000 different random seeds to ensure statistical significance. We compare the mean squared error (MSE) between the true value function and the estimated value function averaged over all states.

The Deterministic Case ($|\mathcal{A}| = 2$, $n = 8$, $k \in \{4, 16, 64\}$, $p_s = 0$, $p_r = 0$) We first examine the effect of k in the simplest setting with a deterministic environment. Since the bootstrapping state is fully deterministic and independent of actions, our analysis (Section 4) suggests that bootstrapping will lose its ability to reduce the variance, and TD would behave like MC asymptotically. Indeed, Figure 4 (top) shows that TD learning, independent of k, approaches the same MSE as MC. Note that this cannot be explained by the bound given by Equation 3, which predicts that the asymptotic error would grow as k increases. On the other hand, we see DAE converging to the true value function with rates matching Bellman iterations, indicating its effectiveness in variance reduction.

The Sticky Transition Case ($|\mathcal{A}| = 2$, $n = 8$, $k \in \{4, 16, 64\}$, $p_s = 0.25$, $p_r = 0$) We now examine how stochastic transitions affect the MSE. Figure 4 (middle) shows that, counterintuitively, stochasticity *reduces* the variance of TD. Furthermore, the learning curves now follow the common bias-variance tradeoff intuition of multi-step learning (i.e., larger k learns faster but leads to higher variance and vice versa). We emphasize that this setting effectively differs from the deterministic case only in how the bootstrapping states are sampled. As such, the variance reduction can only be explained by TD learning's ability to effectively aggregate over a larger number of independent trajectories (cf. Figure 1). DAE converges with rates similar to Bellman iterations again, since B^π does not depend on p_s.

So far, both cases have almost full coverage of the state-action space from the data in each phase, and have variances that can be fully explained by the advantage function (cf. Equation 11). This

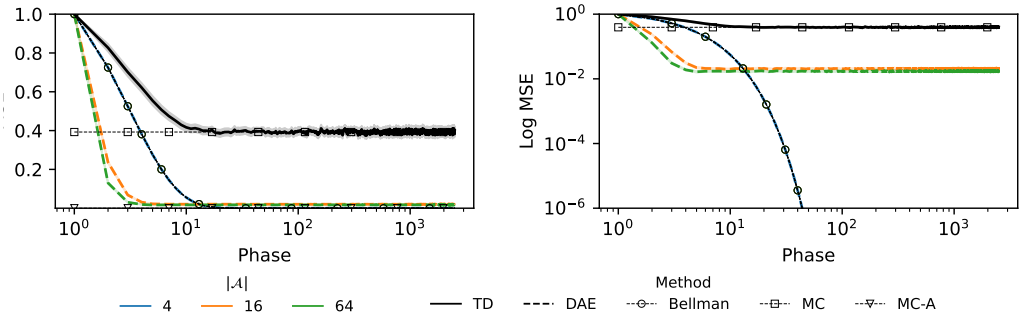


Figure 5: The low coverage case. Lines and shadings represent (mean $\pm$ 3 standard error). Note that TD does not depend on $|\mathcal{A}|$.

allowed DAE to converge almost as fast as Bellman iterations. To see when this breaks down, we next consider the stochastic reward ($B^\pi \not\equiv 0$) and the large $|\mathcal{A}|$ settings.

The Reward Masking Case ($|\mathcal{A}| = 2$, $n = 8$, $k \in \{4, 16, 64\}$, $p_s = 0$, $p_r = 0.2$) Similar to the deterministic case, Figure 4 (bottom) shows that, asymptotically, TD performs similar to MC irrespective of k, as the bootstrapping states are deterministic. On the other hand, the variance of MC-A is no longer zero since $B^\pi \not\equiv 0$, and DAE converges slightly above MC-A, indicating that the finite-sample bias introduced by control variate regression is no longer negligible.

The Low Coverage Case ($|\mathcal{A}| \in \{4, 16, 64\}$, $n = 8$, $k = 16$, $p_s = 0$, $p_r = 0$) Finally, we consider the low coverage case, where the least squares become underdetermined. In the context of RL, this means that most of the actions are not sampled, and the advantage estimates become unreliable. We follow the common practice of choosing the minimum Euclidean-norm solution, as implemented by popular least-squares solvers [Anderson et al., 1999]. Figure 5 shows that, when $|\mathcal{A}| \geq 16$, DAE converges to suboptimal solutions compared to the full coverage cases. Nonetheless, we find that DAE converges to a lower MSE than TD, suggesting that the advantage estimates may help reduce variance even when they are poorly estimated. [1]

8 Related Work

The bias-variance tradeoff between MC and TD has been discussed in classical texts such as Sutton et al. [1998] or Szepesvari [2010]; however, only case studies were given, and a rigorous analysis remained desirable. Kearns and Singh [2000] analyzed the error bounds of multi-step TD learning in the phased setting [Kearns and Singh, 1998], which also serves as the basis of the present work. In the batch setting, Grunewalder et al. [2007] showed that LSTD [Bradtke and Barto, 1996] is statistically more efficient than MC when the Markovian structure of the environment can provide additional information. More recently, Cheikhi and Russo [2023] derived a more precise statistical relationship between batch TD and MC by analyzing trajectory pooling. However, the batch setting abstracts away the iterative nature of TD, and their implications in the online setting remain unclear. Similar problems have also been studied in function approximation settings [Dalal et al., 2018, Bhandari et al., 2018], where finite-sample (time) error bounds were given.

The advantage function [Baird, 1995] is commonly used as control variates for policy gradient methods [Sutton et al., 1999, Greensmith et al., 2004].The present work demonstrates that the advantage function can also be used as control variates for policy evaluation, and shows that DAE [Pan et al., 2022] can be seen as regression adjusted control variates.

9 Discussion

We analyzed the asymptotic behaviors of MC, TD, and DAE, and revealed one of the mechanisms behind the variance reduction property of bootstrapping, namely, the ability to aggregate over a larger

[1] We find DAE converging to the true value function again if one uses iterative solvers (see Appendix B.1).

number of independent trajectories. Furthermore, we established a connection between DAE and control variate regression, and demonstrated how it can further reduce the variance of TD learning. At its core, DAE exploits our knowledge of the policy to reduce variances and an interesting future direction would be to explore other types of control variates for policy evaluation.

Finally, we note some limitations: (1) The phased setting is highly simplified by assuming the ability to sample trajectories uniformly from each state. It remains unclear how we can analyze DAE in an iterative setting, and to what extent the results hold under more realistic sampling (e.g., Markovian sampling) settings. (2) We focused mainly on the variance reduction property of DAE, but it should be noted that DAE also incurs additional space (to store $\mathbf{M}$) and time (to solve linear least-squares) complexities. (3) The theoretical results for DAE only hold up to the first-order approximation case where the control variate coefficients are optimal, and while empirical results seem to suggest that the variance reduction is beneficial, a more rigorous analysis remains desirable.

References

E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999. ISBN 0-89871-447-8 (paperback).

S. Asmussen and P. W. Glynn. *Stochastic simulation: algorithms and analysis*, volume 57. Springer, 2007.

L. Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pages 30–37. Elsevier, 1995.

J. Bhandari, D. Russo, and R. Singal. A finite time analysis of temporal difference learning with linear function approximation. In *Conference on learning theory*, pages 1691–1692. PMLR, 2018.

J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/jax-ml/jax`.

S. J. Bradtke and A. G. Barto. Linear least-squares algorithms for temporal difference learning. *Machine learning*, 22(1):33–57, 1996.

R. M. Burton and U. Rösler. An l2 convergence theorem for random affine mappings. *Journal of applied probability*, 32(1):183–192, 1995.

D. Cheikhi and D. Russo. On the statistical benefits of temporal difference learning. In *International Conference on Machine Learning*, pages 4269–4293. PMLR, 2023.

G. Dalal, B. Szörényi, G. Thoppe, and S. Mannor. Finite sample analyses for td (0) with function approximation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

E. Greensmith, P. L. Bartlett, and J. Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(9), 2004.

S. Grunewalder, S. Hochreiter, and K. Obermayer. Optimality of lstd and its relation to mc. In *2007 International Joint Conference on Neural Networks*, pages 338–343. IEEE, 2007.

C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020. doi: 10.1038/s41586-020-2649-2. URL `https://doi.org/10.1038/s41586-020-2649-2`.

M. Kearns and S. Singh. Finite-sample convergence rates for q-learning and indirect algorithms. *Advances in neural information processing systems*, 11, 1998.

M. J. Kearns and S. Singh. Bias-variance error bounds for temporal difference updates. In *COLT*, pages 142–147, 2000.

A. B. Owen. *Monte Carlo theory, methods and examples*. https://artowen.su.domains/mc/, 2013.

C. C. Paige and M. A. Saunders. Lsqr: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software (TOMS)*, 8(1):43–71, 1982.

H.-R. Pan and B. Schölkopf. Skill or luck? return decomposition via advantage functions. *arXiv preprint arXiv:2402.12874*, 2024.

H.-R. Pan, N. Gürtler, A. Neitz, and B. Schölkopf. Direct advantage estimation. *Advances in Neural Information Processing Systems*, 35:11869–11880, 2022.

M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

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

R. S. Sutton, A. G. Barto, et al. Introduction to reinforcement learning. 1998.

R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

C. Szepesvari. *Algorithms for Reinforcement Learning*. Morgan & Claypool Publishers, 2010.

P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17: 261–272, 2020. doi: 10.1038/s41592-019-0686-2.

A Proofs

A.1 Proof of Corollary 1

Theorem 1. $\lim_{T\to\infty} \mathrm{Var}(V^T_{\mathrm{TD}(k)}(s)) \leq \mathrm{Var}(V_{\mathrm{MC}}(s))$.

Proof. First, we show that $\lim_{T\to\infty} \mathrm{Var}(V^T_{\mathrm{TD}(k)}(s))$ converges. Note that

$$V^T_{\mathrm{TD}(k)}(s) = \bar{\mathbf{R}}^T + \gamma^k \bar{\mathbf{P}}^T \mathbf{V}^{T-1}_{\mathrm{TD}(k)},$$

is a special case of random affine iterated system of the form:

$$X_t = A_t X_{t-1} + b_t, \tag{15}$$

where (A_t, b_t) are i.i.d. random variables. Furthermore, since $\gamma^k \bar{\mathbf{P}}$ is a contraction respect to $||\cdot||_\infty$ and $\mathbf{R}$ is bounded, we know that the random vector $\mathbf{V}^T_{\mathrm{TD}(k)}$ will converge in distribution with respect to the Wasserstein distance W_∞ [Burton and Rösler, 1995]. Consequently, all finite moments of $\mathbf{V}^T_{\mathrm{TD}(k)}$ also converges as $T \to \infty$.

Next, by Lemma 2, we have:

$$\mathrm{Var}(V^T_{\mathrm{TD}(k)}(s)) \leq \sum_{m=0}^{T-1} \gamma^{2km} \mathbb{E}\left[\mathrm{Var}\left(\bar{\mathbf{R}}^{T-m} + \gamma^k \bar{\mathbf{P}}^{T-m} \bar{\mathbf{V}}^{T-1-m}_{\mathrm{TD}(k)} \middle| s_{km} \right) \middle| s_0{=}s \right].$$

It is enough to show that the summation on the right hand side converges to $\mathrm{Var}(V_{\mathrm{MC}}(s))$ as $T \to \infty$. Let $x_{T-m,m} = \mathbb{E}\left[\mathrm{Var}\left(\bar{\mathbf{R}}^{T-m} + \gamma^k \bar{\mathbf{P}}^{T-m} \bar{\mathbf{V}}^{T-1-m}_{\mathrm{TD}(k)} \middle| s_{km} \right) \middle| s_0{=}s \right]$, we are interested in the following limit

$$\lim_{T\to\infty} \sum_{m=0}^{T-1} \gamma^{2km} x_{T-m,m}.$$

Since $x_{\infty,m} \coloneqq \lim_{T\to\infty} x_{T-m,m} = \mathbb{E}\left[\mathrm{Var}\left(\bar{\mathbf{R}} + \gamma^k \bar{\mathbf{P}} V^\pi \middle| s_{km}\right) \middle| s_0{=}s\right]$ (note that both $\mathbf{R}^T$ and $\mathbf{P}^T$ are i.i.d. for all T), there exists $N \in \mathbb{N}$ such that if $T - m > N$ then $|x_{T-m,m} - x_{\infty,m}| < \epsilon$. In addition, since $\mathbf{R}$, $\mathbf{P}$ and $\mathbf{V}^\pi$ are all bounded, there exists $M \in \mathbb{R}$ such that $|x_{\infty,m}| < M$ and $|x_{T-m,m}| < M$ for all $m, T - m \in \mathbb{Z}_+$. Consequently,

$$
\begin{aligned}
&\left|\sum_{m=0}^{T-1} \gamma^{2km} x_{T-m,m} - \sum_{m=0}^{\infty} \gamma^{2km} x_{\infty,m}\right| \\
\leq &\left|\sum_{m=0}^{T-1} \gamma^{2km}(x_{T-m,m} - x_{\infty,m})\right| + \left|\sum_{m=T}^{\infty} \gamma^{2km} x_{\infty,m}\right| \\
\leq &\left|\sum_{m=0}^{T-n-1} \gamma^{2km}(x_{T-m,m} - x_{\infty,m})\right| + \left|\sum_{m=T-n}^{T-1} \gamma^{2km}(x_{T-m,m} - x_{\infty,m})\right| + \frac{M\gamma^{2kT}}{1-\gamma^{2k}} \\
\leq &\frac{\epsilon}{1-\gamma^{2k}} + \frac{2M\gamma^{2k(T-n)}}{1-\gamma^{2k}} + \frac{M\gamma^{2kT}}{1-\gamma^{2k}}
\end{aligned}
$$

is arbitrarily small as $T \to \infty$, and

$$
\lim_{T\to\infty} \mathrm{Var}(V^T_{\mathrm{TD}(k)}(s)) \leq \lim_{T\to\infty} \sum_{m=0}^{T-1} \gamma^{2km} x_{T-m,m} = \sum_{m=0}^{\infty} \gamma^{2km} x_{\infty,m} = \mathrm{Var}(V_{\mathrm{MC}}(s)).
$$

□

A.2 Proof of Corollary 2

Theorem 2. $\limsup_{T\to\infty} \mathrm{Var}(V^{*T}_{\mathrm{DAE}}(s)) \leq \mathrm{Var}(V_{\mathrm{MC-A}}(s)) \leq \mathrm{Var}(V_{\mathrm{MC}}(s))$.

Proof. Since A^π is the optimal control variate for MC estimation, $\mathrm{Var}(V_{\mathrm{MC-A}}(s)) \leq \mathrm{Var} V_{\mathrm{MC}}(s)$ holds. Next, note that $V^{*T}_{\mathrm{DAE}}(s)$ is DAE with the optimal control variate coefficients A^{*T}. Consequently, we must have:

$$
\mathrm{Var}(V^{*T}_{\mathrm{DAE}}(s)) \leq \mathrm{Var}\left(\bar{\mathbf{R}}^T + \gamma^k \bar{\mathbf{P}}^T \mathbf{V}^{*T-1}_{\mathrm{DAE}} - \bar{\mathbf{M}}^T \mathbf{A}^\pi\right), \tag{16}
$$

where $\mathbf{A}^\pi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ is the true advantage function. Since $\mathbf{A}^\pi$ is a constant vector, and $\mathbf{M}^T$ is uniformly bounded, we can rewrite the right hand side of this inequality by

$$
\mathrm{Var}\left(\bar{\mathbf{R}}'^T + \gamma^k \bar{\mathbf{P}}^T \mathbf{V}^{*T-1}_{\mathrm{DAE}} \middle| s_0 = s\right), \tag{17}
$$

where $\mathbf{R}'^T = \mathbf{R}^T - \bar{\mathbf{M}}^T \mathbf{A}^\pi$. By Corollay 1, we know that this variance is bounded above by the variance of the MC estimator with this new reward function, which is precisely the variance of $V_{\mathrm{MC-A}}$. □

Finally, we make a remark about this corollary. Since the advantage estimate $\mathbf{A}^T$ now also depends on $\mathbf{V}^{T-1}$, it is not clear whether the update remains a contraction, or whether higher moments (e.g., variance) also converge. As such, we only prove the supremum limit is upper bounded by $V_{\mathrm{MC-A}}$.

B Experimental Details

Algorithm 1 shows the pseudocode. All experiments are based on Python with least-square solvers implemented by `NumPy` [Harris et al., 2020], `SciPy` [Virtanen et al., 2020] or `JAX` [Bradbury et al., 2018]. A single run (1 seed, 2500 phases) takes less than a minute on commercial CPUs, except for the large $|\mathcal{A}|$ experiment, where we leveraged GPUs (Nvidia A100) to parallelize the least-square solver. We use LSQR [Paige and Saunders, 1982] as the default least-square solver as we found it to be slightly faster. The only exception is the large $|\mathcal{A}|$ experiment, where we used the SVD-based minimum norm solver [Anderson et al., 1999] to ensure reproducibility.

Algorithm 1 Phased TD/DAE

Require: n, k, `alg`$\in$\{TD, DAE\}, `LSTSQ_SOLVER`

Initialize $\mathbf{V} \equiv 0$
for $T = 1, 2, \ldots$ **do**
  $\mathcal{D} = \{\}$
  for $s \in \mathcal{S}$ **do**
    for $i = 1, \ldots, n$ **do**
      Sample k-step trajectory τ from environment
      $\mathcal{D} \leftarrow \mathcal{D} \cup \{\tau\}$
    end for
  end for
  if `alg` == TD **then**
    Compute $\bar{\mathbf{R}}$, $\bar{\mathbf{P}}$ from $\mathcal{D}$
    $\mathbf{V} \leftarrow \bar{\mathbf{R}} + \gamma^k \bar{\mathbf{P}} \mathbf{V}$
  else
    Compute $\bar{\mathbf{M}}$, $\bar{\mathbf{R}}$, $\bar{\mathbf{P}}$ from $\mathcal{D}$
    $\mathbf{V}, \mathbf{A} \leftarrow$ `LSTSQ_SOLVER`$(||\bar{\mathbf{R}} + \gamma^k \bar{\mathbf{P}} \mathbf{V} - \hat{\mathbf{V}} - \bar{\mathbf{M}} \hat{\mathbf{A}}||^2)$
  end if
end for

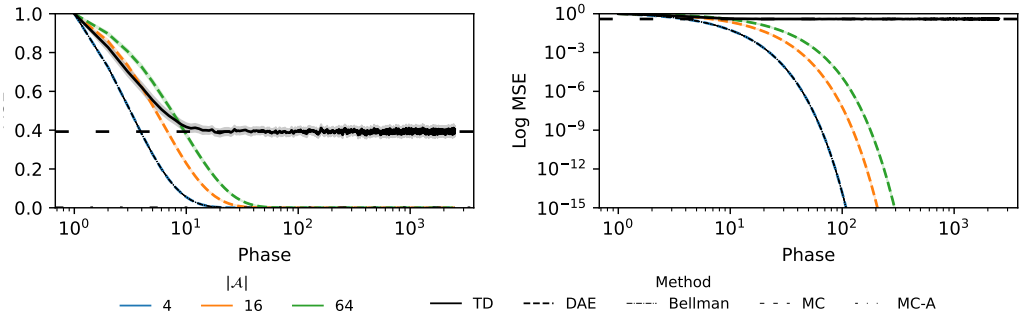


Figure 6: The low coverage case with an iterative solver. Lines and shadings represent (mean $\pm$ 3 standard error). Note that TD does not depend on $|\mathcal{A}|$.

B.1 Additional Experiments

The Low Coverage Case With an Iterative Solver ($|\mathcal{A}| \in \{4, 16, 64\}$, $n = 8$, $k = 16$, $p_s = 0$, $p_r = 0$) In Section 7, we showed that increasing the size of the action space results in DAE converging to suboptimal solutions when regularized with minimum norm solutions. In Figure 6, we rerun the same experiment but with an iterative least-sqaures solver (LSQR [Paige and Saunders, 1982] in this case), where the optimum in the previous phase is used as the initialization for the current phase. We find that DAE converges again to the true value function, although at a slower rate as $|\mathcal{A}|$ increases. This might partially explain the success of DAE in the deep RL setting [Pan et al., 2022], where gradient-based optimization is used.