A ADDITIONAL RESULTS

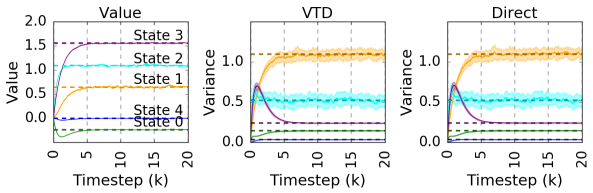

Figure 13: **Complex MDP** evaluated on-policy with all step-sizes equal ($\alpha = \bar{\alpha} = 0.01$). Both algorithms achieve similar results.

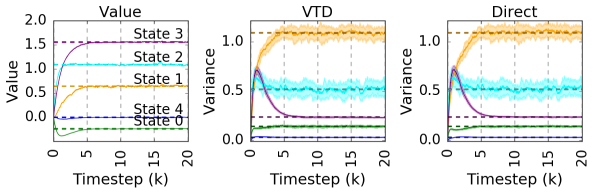

Figure 14: **Complex MDP**. Using traces in the secondary estimators, but not in the value estimator (TD(λ), $\alpha = \bar{\alpha} = 0.01$, $\kappa = 0.0, \bar{\kappa} = 1.0$). There is no significant difference in performance between DVTD and VTD.

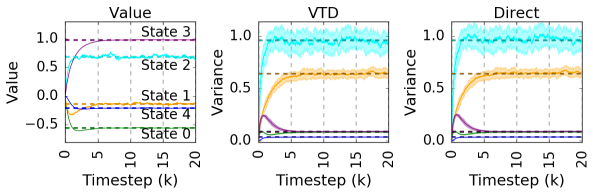

Figure 15: **Complex MDP** estimating V from off-policy samples ($\alpha = \bar{\alpha} = 0.01, \eta = 1, \bar{\rho} = \rho$). Both methods produce similar results.

B EXTENDED DERIVATION

Lemma 1. *For $j(s) = \mathbb{E}\left[G^\lambda_{t+1} | S_t = s\right]$, i.e., satisfying the Bellman function $b : \mathcal{S} \times \mathcal{A} \times \mathbb{R} \times \mathcal{S} \to \mathbb{R}$,*

$$\mathbb{E}[b(S_t, A_t, R_{t+1}, S_{t+1})(G^\lambda_{t+1} - j(S_{t+1}))|S_t = s] = 0$$

Proof. Let $b_t = b(S_t, A_t, R_{t+1}, S_{t+1})$. By the law of total expectation:

$$\begin{aligned}
&\mathbb{E}\left[b_t(G^\lambda_{t+1} - j(S_{t+1}))|S_t = s\right] \\
&= \mathbb{E}\left[\mathbb{E}[b_t(G^\lambda_{t+1} - j(S_{t+1}))|S_t, A_t, S_{t+1}]|S_t = s\right]
\end{aligned}$$

Given S_t, A_t, R_{t+1} and S_{t+1}, b_t is constant and can be moved outside of the expectation. Therefore,

$$\begin{aligned}
&\mathbb{E}[b_t(G^\lambda_{t+1} - j(S_{t+1}))\Big|S_t, A_t, R_{t+1}, S_{t+1}] \\
&= \mathbb{E}\left[b_t\middle|S_t, A_t, R_{t+1}, S_{t+1}\right] \\
&\quad \times \mathbb{E}\left[G^\lambda_{t+1} - j(S_{t+1})\middle|S_t, A_t, R_{t+1}, S_{t+1}\right]
\end{aligned}$$

Because

$$\mathbb{E}\left[G^\lambda_{t+1} - j(S_{t+1})\middle|S_t, A_t, R_{t+1}, S_{t+1}\right] = 0$$

the result follows. □

Theorem 3.

$$\begin{aligned}
v(s) = \mathbb{E}[&(\eta_t\delta_t + (\eta_t - 1)j(s))^2 \\
&+ \lambda^2_{t+1}\gamma^2_{t+1}\eta^2_t v(S_{t+1})|S_t = s]
\end{aligned}$$

Proof. The proof is similar to the proof of Theorem 1.

$$\begin{aligned}
&v(s) = \mathbb{E}[\{G^\lambda_t - j(S_t)\}^2|S_t = s] \\
=&\mathbb{E}[\{\eta_t R_{t+1} + \eta_t\gamma_{t+1}(1 - \lambda_{t+1})j(S_{t+1}) \\
&+ \eta_t\gamma_{t+1}\lambda_{t+1}G^\lambda_{t+1} - j(s)\}^2|S_t = s] \\
=&\mathbb{E}[\{\eta_t R_{t+1} + \eta_t\gamma_{t+1}j(S_{t+1}) - \eta_t j(s) + \eta_t j(s) \\
&- \eta_t\gamma_{t+1}\lambda_{t+1}j(S_{t+1}) \\
&+ \eta_t\gamma_{t+1}\lambda_{t+1}G^\lambda_{t+1} - j(s)\}^2|S_t = s] \\
=&\mathbb{E}[\{(\eta_t\delta_t + (\eta_t - 1)j(s)) \\
&+ \eta_t\gamma_{t+1}\lambda_{t+1}(G^\lambda_{t+1} - j(S_{t+1}))\}^2|S_t = s] \\
=&\mathbb{E}[(\eta_t\delta_t + (\eta_t - 1)j(s))^2 \\
&+ \eta^2_t\gamma^2_{t+1}\lambda^2_{t+1}(G^\lambda_{t+1} - j(S_{t+1}))^2 \\
&+ 2\eta_t\gamma_{t+1}\lambda_{t+1}(\eta_t\delta_t + (\eta_t - 1)j(s)) \\
&\quad (G^\lambda_{t+1} - j(S_{t+1}))|S_t = s] \\
=&\mathbb{E}[(\eta_t\delta_t + (\eta_t - 1)j(s))^2 \\
&+ \eta^2_t\gamma^2_{t+1}\lambda^2_{t+1}(G^\lambda_{t+1} - j(S_{t+1}))^2 \\
&+ 2\eta^2_t\gamma_{t+1}\lambda_{t+1}\delta_t(G^\lambda_{t+1} - j(S_{t+1})) \\
&+ 2\eta_t\gamma_{t+1}\lambda_{t+1}(\eta_t - 1)j(s)(G^\lambda_{t+1} - j(S_{t+1}))|S_t = s]
\end{aligned} \tag{17}$$

Using Lemma 1, with different fixed functions b, we can conclude that the last two terms are zero, giving

$$\begin{aligned}
&v(s) = \mathbb{E}[(\eta_t\delta_t + (\eta_t - 1)j(s))^2 \\
&+ \eta^2_t\gamma^2_{t+1}\lambda^2_{t+1}(G^\lambda_{t+1} - j(S_{t+1}))^2|S_t = s]
\end{aligned}$$

By the law of total expectation

$$\begin{aligned}
&v(s) = \mathbb{E}[(\eta_t\delta_t + (\eta_t - 1)j(s))^2 \\
&+ \mathbb{E}[\eta^2_t\gamma^2_{t+1}\lambda^2_{t+1}(G^\lambda_{t+1} - j(s'))^2|S_{t+1} = s']|S_t = s] \\
&= \mathbb{E}[(\eta_t\delta_t + (\eta_t - 1)j(s))^2 \\
&+ \eta^2_t\gamma^2_{t+1}\lambda^2_{t+1}v(S_{t+1})|S_t = s].
\end{aligned}$$

completing the proof. □

Theorem 3 gives a Bellman equation for $V(s)$ in the more general off-policy setting. The resulting TD algorithm uses meta-reward $(\eta_t\delta_t + (\eta_t - 1)j(s))^2$ and discounting function $\eta_t^2\gamma_{t+1}^2\lambda^2$.

C FUNCTION APPROXIMATION DETAILS

In Section 4.6 we showed our results on the domain shown in Figure 12(a). This domain was previously investigated by Tamar et al. (2016) for indirectly estimating the variance of the return with LSTD(λ) using $\bar{\kappa} = 0.95$. We define the RMSVE for a value and variance estimate on this domain to be

$$\text{RMSVE}(J) = \sqrt{\sum_{i=0}^{29} d_\pi(s_i)(J(s_i) - j(s_i))^2}$$

$$\text{RMSVE}(V) = \sqrt{\sum_{i=0}^{29} d_\pi(s_i)(V(s_i) - v(s_i))^2}$$

where $d_\pi(s_i)$ is the steady state probability of being in state s_i.

For the linear function approximation setting each of our estimators is simply an inner product of a set of weights and a feature vector: $J(s) = \mathbf{w}_J^\top \phi_J(s)$, $M(s) = \mathbf{w}_M^\top \phi_M(s)$, and $V(s) = \mathbf{w}_V^\top \phi_V(s)$. DVTD with linear function approximation and accumulating traces is given by modifying Algorithm 15:

DVTD with Linear Function Approximation

$$\begin{aligned}
\bar{R}_{t+1} &\leftarrow (\eta_t\delta_t + (\eta_t - 1)J_{t+1}(s))^2 \\
\bar{\gamma}_{t+1} &\leftarrow \gamma_{t+1}^2\lambda_{t+1}^2\eta_t^2 \\
\bar{\delta}_t &\leftarrow \bar{R}_{t+1} + \bar{\gamma}_{t+1}V_t(s') - V_t(s) \\
\bar{\mathbf{e}}_t &\leftarrow \bar{\rho}_t(\bar{\gamma}_t\bar{\kappa}_t\bar{\mathbf{e}}_{t-1} + \phi_t(S_t)) \\
\mathbf{w}_{V:t+1} &\leftarrow \mathbf{w}_{V:t} + \bar{\alpha}\bar{\delta}_t\bar{\mathbf{e}}_t
\end{aligned} \tag{18}$$

For DVTD, variance is computed directly as $V_{t+1}(s)$.

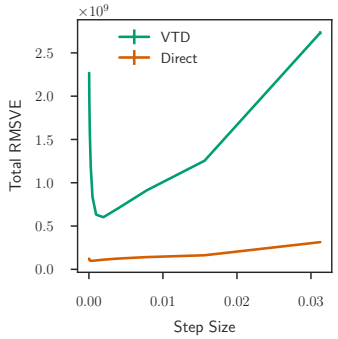

Figure 16: **Random Walk**. Sensitivity to the step-size for VTD and DVTD. Error bars are shown but almost invisible.

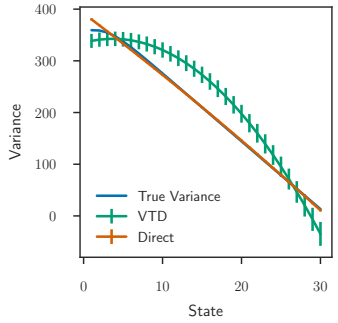

Figure 17: **Random Walk**. Average final estimates of variance reached after 3,000,000 timesteps with 100 runs.

VTD with linear function approximation is given as:

VTD with Linear Function Approximation

$$\begin{aligned}
\bar{G}_t &\leftarrow R_{t+1} + \gamma_{t+1}(1 - \lambda_{t+1})J_{t+1}(s') \\
\bar{R}_{t+1} &\leftarrow \eta_t^2\bar{G}_t^2 + 2\eta_t^2\gamma_{t+1}\lambda_{t+1}\bar{G}_t J_{t+1}(s') \\
\bar{\gamma}_{t+1} &\leftarrow \eta_t^2\gamma_{t+1}^2\lambda_{t+1}^2 \\
\bar{\delta}_t &\leftarrow \bar{R}_{t+1} + \bar{\gamma}_{t+1}M_t(s') - M_t(s) \\
\bar{\mathbf{e}}_t &\leftarrow \bar{\rho}_t(\bar{\gamma}_t\bar{\kappa}_t\bar{\mathbf{e}}_{t-1} + \phi_t(S_t)) \\
\mathbf{w}_{M:t+1}(s) &\leftarrow \mathbf{w}_{M:t}(s) + \bar{\alpha}\bar{\delta}_t\bar{\mathbf{e}}_t(s)
\end{aligned} \tag{19}$$

For VTD, variance is computed as $V_{t+1}(s) = M_{t+1}(s) - J_{t+1}(s)^2$.

While we previously reported our step-size selection strategy in Section 4.6, in Figure 16 we show the total RMSVE over 3,000,000 timesteps as a function of the step-size for the variance learner with the step-size for the value learner the same as in Section 4.6. Note that we only show step-sizes that did not lead to numerical errors and each of the values reported is the mean of 30 runs. Here we see that the direct method was con-

siderably more insensitive to the step-size selection than VTD.

After 3,000,000 timesteps, and using the best alpha for VTD from the sweep shown in Figure 16 ($\alpha_J = 2^{-11} \approx 0.0005$, $\alpha_M = \alpha_V = 2^{-9} \approx 0.002$), we obtain the estimates shown in Figure 17. Note that each of the values reported are the mean of 100 runs.

D ADADELTA STEP-SIZES

The step-sizes generated by the ADADELTA algorithm in Figure 7 are shown in Figure 18. As we evaluate in the tabular case at each timestep, only the step-size for the current state has any impact. Thus, the values shown here are the average step-size used over each episode.

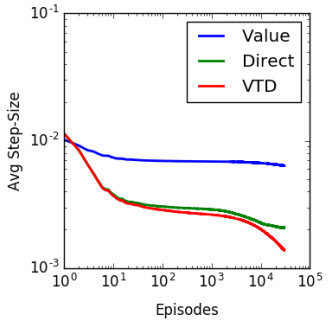

Figure 18: **Chain MDP**. The average step-sizes computed by ADADELTA in Figure 7.

E VARIABILITY IN UPDATES

In this section, we show the effective update to $V_t(s)$ on each timestep for each of the two algorithms in the on-policy setting. For notational clarity let $r = r_{t+1}$, $\alpha = \alpha_t$, $\gamma = \gamma_{t+1}$, $\lambda = \lambda_{t+1}$, $s = s_t$, $s' = s_{t+1}$, and $\delta_t = \delta$.

For the direct algorithm the change is just:

$$\Delta V_t(s) = \bar{\alpha}(\delta^2 + \bar{\gamma} V_t(s') - V_t(s)). \tag{20}$$

The updates for the VTD algorithm are much more complicated to compute so we will make some assumptions about the domain to simplify the derivation. First, we compute the change in the second moment and value estimators separately.

We first expand the term δ^2 as follows:

$$\begin{aligned}
\delta =& r + \gamma J_t(s') - J_t(s) \\
\delta^2 =& (r + \gamma J_t(s'))^2 \\
& - 2(r + \gamma J_t(s')) J_t(s) + J_t(s)^2.
\end{aligned}$$

Now we expand the change in the second moment estimate, M. To simplify the expansion, we assume that at each transition the agent moves to a new state, i.e., $s_t \neq s_{t+1} \forall t$ (this is not required for our algorithm, but simplifies the expansions below). This assumption holds for both of the tabular domains examined in this paper. This allows us to substitute $J_{t+1}(s') = J_t(s)$, which greatly simplifies the updates.

$$\begin{aligned}
\Delta M(s) =& \bar{\alpha}[(r + \gamma J_{t+1}(s'))^2 \\
& - \bar{\gamma}^2 J_{t+1}(s')^2 + \bar{\gamma} M_t(s') - M_t(s)] \\
=& \bar{\alpha}[(r + \gamma J_t(s'))^2 \\
& - \bar{\gamma}^2 J_t(s')^2 + \bar{\gamma} M_t(s') - M_t(s)] \\
=& \bar{\alpha}[(r + \gamma J_t(s'))^2 - 2(r + \gamma J_t(s')) J_t(s) \\
& + J_t(s)^2 + 2(r + \gamma J_t(s')) J_t(s) - J_t(s)^2 \\
& - \bar{\gamma}^2 J_t(s')^2 + \bar{\gamma} M_t(s') - M_t(s)] \\
=& \bar{\alpha}[\delta^2 + 2(r + \gamma J_t(s')) J_t(s) - J_t(s)^2 \\
& - \bar{\gamma}^2 J_t(s')^2 + \bar{\gamma} M_t(s') - M_t(s)]
\end{aligned}$$

Notice that from the definition of the TD error: $R + \gamma J_t(s') = \delta + J_t(s)$.

$$\begin{aligned}
=& \bar{\alpha}[\delta^2 + 2(\delta + J_t(s)) J_t(s) - J_t(s)^2 \\
& - \bar{\gamma}^2 J_t(s')^2 + \bar{\gamma} M_t(s') - M_t(s)] \\
=& \bar{\alpha}[\delta^2 + 2\delta J_t(s) + J_t(s)^2 - \bar{\gamma}^2 J_t(s')^2 \\
& + \bar{\gamma} M_t(s') - M_t(s)] \\
=& \bar{\alpha}[\delta^2 + (\bar{\gamma} M_t(s') - \bar{\gamma} J_t(s')^2) \\
& - (M_t(s) - J_t(s)^2) + 2\delta J_t(s) - \bar{\gamma}^2 J_t(s')^2 \\
& + \bar{\gamma} J_t(s')^2] \\
=& \bar{\alpha}[\delta^2 + \bar{\gamma} V_t(s') - V_t(s) + 2\delta J_t(s) \\
& - \bar{\gamma}^2 J_t(s')^2 + \bar{\gamma} J_t(s')^2] \\
=& \bar{\alpha}[\delta^2 + \bar{\gamma} V_t(s') - V_t(s)] \\
& + \bar{\alpha}[2\delta J_t(s) - \bar{\gamma}^2 J_t(s')^2 + \bar{\gamma} J_t(s')^2]
\end{aligned}$$

The first half of this equation is the same as the update for the direct algorithm (20). Now we expand the change in the variance update for VTD.

$$\begin{aligned}
\Delta V_t(s) =& (M_{t+1}(s) - J_{t+1}(s)^2) - (M_t(s) - J_t(s)^2) \\
=& \Delta M(s) + J_t(s)^2 - J_{t+1}(s)^2 \\
=& \Delta M(s) + J_t(s)^2 - (\alpha\delta + J_t(s))^2 \\
=& \Delta M(s) + J_t(s)^2 \\
& - ((\alpha\delta)^2 + 2\alpha\delta J_t(s) + J_t(s)^2) \\
=& \Delta M(s) - (\alpha\delta)^2 - 2\alpha\delta J_t(s).
\end{aligned}$$

Table 2: Average updates for various experiments.

Fig.	Value	2nd Moment	VTD	Direct
4(a)	0.00332	0.0157	0.00415	0.00415
4(b)	0.0322	0.0165	0.143	0.00387
4(c)	0.00332	0.156	0.142	0.0419
5	0.0	0.0166	0.0166	0.00381
7	0.0212	0.0306	0.0752	0.00884
13	0.00362	0.00675	0.00381	0.00385
15	0.00362	0.00461	0.00303	0.00307
11	0.00362	0.0110	0.0116	0.00838

Note that in the case that $\alpha = \bar{\alpha}$ this last term cancels out and we're left with:

$$\begin{aligned}\Delta V_t(s) =& \alpha[\delta^2 + \bar{\gamma} V_t(s') - V_t(s)] + \\ & \alpha J_t(s')^2(\bar{\gamma} - \bar{\gamma}^2) - (\alpha\delta)^2.\end{aligned}$$

This suggests that VTD will deviate from the direct method more when: α is larger, $J_t(s')$ is larger, $\bar{\gamma} = 0.5$ and for large values of δ. In general, we expect from this equation that the updates for the VTD will be larger than those of the direct method, suggesting a cause for the higher variance of variance estimates across runs as observed for VTD under a number of scenarios.

We also empirically tested this hypothesis, with Table 2 showing the updates for the two algorithms across the tabular experiments. For episodic tasks (chain MDP, Figures 4(a)-7) the results show the average total absolute change over all states for a given episode averaged across runs and then averaged across all episodes. For the continuing case (complex MDP, Figures 13-11) the results are the average absolute change for a timestep averaged over all runs and then averaged across the entire run length. The experiments shaded in gray are those where the two algorithms behaved nearly identically. In this case, we see that the average magnitude of updates is nearly identical. For the other experiments, the VTD algorithm showed higher variance in its variance estimates across runs. For these experiments, we see that the average magnitude of the VTD updates is much larger than for the direct algorithm.