Loosely Consistent Emphatic Temporal-Difference Learning (Supplementary Material)

Jiamin He[1] **Fengdi Che**[1] **Yi Wan**[1] **A. Rupam Mahmood**[1,2]

[1]Department of Computing Science, University of Alberta
[2] CIFAR AI Chair, Alberta Machine Intelligence Institute (Amii)

A PROOF OF THE CONSISTENCY OF AETD AND LC-ETD

Since AETD(0) is a special case of LC-ETD(λ, β, ν) with $\lambda = 0$, $\beta = 0$, and $\nu = 1$, the proof for Theorem 3.1 is also a special case of the proof for Theorem 3.2, which will be presented below.

We first revisit the update of LC-ETD(λ, β, ν):

$$\begin{aligned}
\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \alpha\delta_t \mathbf{z}_t, \\
\delta_t &= R_{t+1} + \gamma \boldsymbol{\phi}_{t+1}^\top \boldsymbol{\theta}_t - \boldsymbol{\phi}_t^\top \boldsymbol{\theta}_t, \\
\mathbf{z}_t &= \rho_t(\gamma\lambda \mathbf{z}_{t-1} + M_t \boldsymbol{\phi}_t), \text{with } \mathbf{z}_{-1} = \mathbf{0}, \\
M_t &= (1 - \lambda h(t))\, F_t + \lambda g(t), \\
F_t &= (1 - g(t))\, \rho_{t-1} F_{t-1} + g(t), \text{with } F_0 = 1,
\end{aligned} \tag{A.1}$$

where $h(t)$ and $g(t)$ are defined as follows:

$$h(t) \doteq \left(\frac{1-\beta}{t+1}\right)^\nu \text{ and } g(t) \doteq \frac{1-\beta}{(t+1)^\nu}$$

with $\beta \in [0, 1)$ and $\nu \in (0, 1]$, or $\beta = 1$ and $\nu \in [0, 1]$. Then we present the relationship between F_t, M_t, and the density ratio with Lemma A.1 and Lemma A.2.

Lemma A.1. *Under Assumption 2.1 and 2.2, for any $\beta \in [0, 1)$ and $\nu \in (0, 1]$, or $\beta = 1$ and $\nu \in [0, 1]$, if $\lim_{t\to\infty} \mathbb{E}_\mu[F_t|S_t = s]$ exists for all $s \in \mathcal{S}$, where F_t is defined in Update (A.1), then*

$$\lim_{t\to\infty} \mathbb{E}_\mu[F_t|S_t = s] = \frac{d_\pi(s)}{d_\mu(s)}$$

holds for any $s \in \mathcal{S}$.

Proof. Let $\mathbf{f} = [f(s_1), \cdots, f(s_{|\mathcal{S}|})]^\top \in \mathbb{R}^{|\mathcal{S}|}$, and $f(s) \in \mathbb{R}$ is defined as follows:

$$f(s) \doteq d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu[F_t|S_t = s], \text{ for any } s \in \mathcal{S}, \tag{A.2}$$

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

which exists under our assumptions. Then we have

$$
\begin{aligned}
f(s) &= d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu[F_t|S_t = s] \\
&= d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu\left[(1-g(t))\rho_{t-1}F_{t-1} + g(t)|S_t = s\right] \\
&= d_\mu(s) \left(\lim_{t\to\infty}(1-g(t))\mathbb{E}_\mu[\rho_{t-1}F_{t-1}|S_t = s] + \lim_{t\to\infty} g(t)\right) && \text{(A.3)} \\
&= d_\mu(s) \lim_{t\to\infty}(1-g(t)) \lim_{t\to\infty} \mathbb{E}_\mu[\rho_{t-1}F_{t-1}|S_t = s] && \text{(A.4)} \\
&= d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu[\rho_{t-1}F_{t-1}|S_t = s] \\
&= d_\mu(s) \lim_{t\to\infty} \sum_{\bar{s},\bar{a}} \mathbb{P}_\mu(S_{t-1} = \bar{s}, A_{t-1} = \bar{a}|S_t = s) \frac{\pi(\bar{a}|\bar{s})}{\mu(\bar{a}|\bar{s})} \mathbb{E}_\mu[F_{t-1}|S_{t-1} = \bar{s}] \\
&= d_\mu(s) \lim_{t\to\infty} \sum_{\bar{s},\bar{a}} \frac{\mathbb{P}_\mu(S_{t-1} = \bar{s}, A_{t-1} = \bar{a}, S_t = s)}{\mathbb{P}_\mu(S_t = s)} \frac{\pi(\bar{a}|\bar{s})}{\mu(\bar{a}|\bar{s})} \mathbb{E}_\mu[F_{t-1}|S_{t-1} = \bar{s}] \\
&= d_\mu(s) \sum_{\bar{s},\bar{a}} \frac{d_\mu(\bar{s})\mu(\bar{a}|\bar{s})p(s|\bar{s},\bar{a})}{d_\mu(s)} \frac{\pi(\bar{a}|\bar{s})}{\mu(\bar{a}|\bar{s})} \lim_{t\to\infty} \mathbb{E}_\mu[F_{t-1}|S_{t-1} = \bar{s}] \\
&= \sum_{\bar{s},\bar{a}} \pi(\bar{a}|\bar{s})p(s|\bar{s},\bar{a})d_\mu(\bar{s}) \lim_{t\to\infty} \mathbb{E}_\mu[F_{t-1}|S_{t-1} = \bar{s}] \\
&= \sum_{\bar{s}} [\mathbf{P}_\pi]_{\bar{s}s} f(\bar{s}),
\end{aligned}
$$

where in Eqs. (A.3) and (A.4), we use the assumption that $\lim_{t\to\infty} \mathbb{E}_\mu[F_t|S_t = s]$ exists for any $s \in \mathcal{S}$ and the facts that $\lim_{t\to\infty}(1-g(t)) = 1$ and $\lim_{t\to\infty} g(t) = 0$ for any $\beta \in [0,1)$ and $\nu \in (0,1]$, or $\beta = 1$ and $\nu \in [0,1]$. From the last equation, we have $\mathbf{f}^\top = \mathbf{f}^\top \mathbf{P}_\pi$ in vector form. Since the expectations of importance-sampling ratios are one and $F_0 = 1$, by induction, the expectation of F_t will remain one for any $t \in \mathbb{N}$. Then we have:

$$
\begin{aligned}
\mathbf{1}^\top \mathbf{f} = \sum_s f(s) &= \sum_{s\in\mathcal{S}} d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu[F_t|S_t = s] \\
&= \lim_{t\to\infty} \mathbb{E}_\mu[F_t|S_t] = 1.
\end{aligned}
$$

By Assumption 2.1, the existence of the target policy's stationary distribution is unique. From $\mathbf{f}^\top = \mathbf{f}^\top \mathbf{P}_\pi$ and $\mathbf{1}^\top \mathbf{f} = 1$, we can infer that $\mathbf{f} = \mathbf{d}_\pi$, that is,

$$
d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu[F_t|S_t = s] = d_\pi(s). \tag{A.5}
$$

Since it holds that $d_\mu(s) > 0$ for any $s \in \mathcal{S}$ by Assumption 2.1, we can divide both sides of Eq. (A.5) by $d_\mu(s)$ and conclude the proof. □

Lemma A.2. *Under the assumptions of Lemma A.1, for any $\beta \in [0,1)$ and $\nu \in (0,1]$, or $\beta = 1$ and $\nu \in [0,1]$, it holds for any $s \in \mathcal{S}$ that*

$$
\lim_{t\to\infty} \mathbb{E}_\mu[M_t|S_t = s] = \frac{d_\pi(s)}{d_\mu(s)},
$$

where M_t is defined in Update (A.1).

Proof. We can expand M_t and use the result from Lemma A.1:

$$
\begin{aligned}
\lim_{t\to\infty} \mathbb{E}_\mu[M_t|S_t = s] &= \lim_{t\to\infty} \mathbb{E}_\mu[(1-\lambda h(t))F_t + \lambda g(t)|S_t = s] \\
&= \lim_{t\to\infty}(1-\lambda h(t))\mathbb{E}_\mu[F_t|S_t = s] + \lim_{t\to\infty} \lambda g(t) \\
&= \lim_{t\to\infty}(1-\lambda h(t)) \lim_{t\to\infty} \mathbb{E}_\mu[F_t|S_t = s] && \text{(A.6)} \\
&= \lim_{t\to\infty} \mathbb{E}_\mu[F_t|S_t = s] && \text{(A.7)} \\
&= \frac{d_\pi(s)}{d_\mu(s)}, && \text{(Lemma A.1)}
\end{aligned}
$$

where, in Eqs. (A.6) and (A.7), we make use of $\lim_{t\to\infty} g(t) = \lim_{t\to\infty}(1-\beta)(t+1)^{-\nu} = 0$ and $\lim_{t\to\infty} h(t) = \lim_{t\to\infty}(1-\beta)^{\nu}(t+1)^{-\nu} = 0$ for any $\beta \in [0,1)$ and $\nu \in (0,1]$, or $\beta = 1$ and $\nu \in [0,1]$. □

From Lemma A.1 and Lemma A.2, we can see that the expectations of both F_t and M_t converge to the density ratio $\frac{d_\pi(s)}{d_\mu(s)}$. By utilizing these results, we can prove the consistency of LC-ETD(λ, β, ν) (including AETD(λ)), which is presented in Theorem A.3.

Theorem A.3 (Restatement of Theorem 3.2)**.** *Let Assumptions 2.1-2.3 hold. For any $\beta \in [0,1)$ and $\nu \in (0,1]$, or $\beta = 1$ and $\nu \in [0,1]$, if $\lim_{t\to\infty} \mathbb{E}_\mu[F_t|S_t = s]$ and $\lim_{t\to\infty} \mathbb{E}_\mu[\mathbf{z}_t|S_t = s]$ exist for all $s \in \mathcal{S}$, then LC-ETD(λ, β, ν) has the same expected update as On-policy TD(λ). As a result, LC-ETD(λ, β, ν) is stable and consistent.*

Remark A.4. AETD(λ) is stable and consistent, as it is a special case of LC-ETD(λ, β, ν) with $\beta = 0$ and $\nu = 1$.

Proof of Theorem A.3. The proof is similar in structure to the proof of Theorem 1 in the work of Sutton et al. (2016). We start from the update of LC-ETD(λ, β, ν). Specifically, we can rewrite Update (A.1) as follows:

$$\begin{aligned}
\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \alpha\delta_t\mathbf{z}_t \\
&= \boldsymbol{\theta}_t + \alpha\left(R_{t+1} + \gamma\boldsymbol{\phi}_{t+1}^\top\theta_t - \boldsymbol{\phi}_t^\top\theta_t\right)\mathbf{z}_t \\
&= \boldsymbol{\theta}_t + \alpha\left(\underbrace{\left[\mathbf{z}_t R_{t+1}\right]}_{\mathbf{b}_t} - \underbrace{\left[\mathbf{z}_t(\boldsymbol{\phi}_t - \gamma\boldsymbol{\phi}_{t+1})^\top\right]}_{\mathbf{A}_t}\boldsymbol{\theta}_t\right).
\end{aligned} \tag{A.8}$$

Defining $\mathbf{A} \doteq \lim_{t\to\infty} \mathbb{E}_\mu[\mathbf{A}_t]$ and $\mathbf{b} \doteq \lim_{t\to\infty} \mathbb{E}_\mu[\mathbf{b}_t]$, we analyze LC-ETD($\lambda$, β, ν)'s expected update:

$$\bar{\boldsymbol{\theta}}_{t+1} = \bar{\boldsymbol{\theta}}_t + \alpha(\mathbf{b} - \mathbf{A}\bar{\boldsymbol{\theta}}_t). \tag{A.9}$$

We first analyze the $\mathbf{A}$ matrix. Similar to obtain ETD(λ)'s $\mathbf{A}$ matrix (Sutton et al., 2016), we have

$$\begin{aligned}
\mathbf{A} = \lim_{t\to\infty} \mathbb{E}_\mu[\mathbf{A}_t] &= \lim_{t\to\infty} \mathbb{E}_\mu\left[\mathbf{z}_t(\boldsymbol{\phi}_t - \gamma\boldsymbol{\phi}_{t+1})^\top|S_t = s\right] \\
&= \sum_s d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu\left[\mathbf{z}_t(\boldsymbol{\phi}_t - \gamma\boldsymbol{\phi}_{t+1})^\top|S_t = s\right] \\
&= \sum_s d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu\left[\rho_t(\gamma\lambda\mathbf{z}_{t-1} + M_t\boldsymbol{\phi}_t)(\boldsymbol{\phi}_t - \gamma\boldsymbol{\phi}_{t+1})^\top|S_t = s\right] \\
&= \sum_s d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu\left[(\gamma\lambda\mathbf{z}_{t-1} + M_t\boldsymbol{\phi}_t)|S_t = s\right]\mathbb{E}_\mu\left[\rho_t(\boldsymbol{\phi}_t - \gamma\boldsymbol{\phi}_{t+1})^\top|S_t = s\right] \\
&\text{(because, } \gamma\lambda\mathbf{z}_{t-1} + M_t\boldsymbol{\phi}_t \text{ is independent of } \rho_t(\boldsymbol{\phi}_t - \gamma\boldsymbol{\phi}_{t+1})^\top \text{ if } S_t \text{ is given)} \\
&= \sum_s \underbrace{d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu\left[\gamma\lambda\mathbf{z}_{t-1} + M_t\boldsymbol{\phi}_t|S_t = s\right]}_{\mathbf{z}(s)\in\mathbb{R}^d}\mathbb{E}_\mu\left[\rho_k(\boldsymbol{\phi}_k - \gamma\boldsymbol{\phi}_{k+1})^\top|S_k = s\right] \\
&= \sum_s \mathbf{z}(s)\mathbb{E}_\mu\left[\rho_k(\boldsymbol{\phi}_k - \gamma\boldsymbol{\phi}_{k+1})^\top|S_k = s\right] \\
&= \sum_s \mathbf{z}(s)\mathbb{E}_\pi\left[\boldsymbol{\phi}_k - \gamma\boldsymbol{\phi}_{k+1}|S_k = s\right]^\top \\
&= \sum_s \mathbf{z}(s)\left(\boldsymbol{\phi}(s) - \gamma\sum_{s'}[\mathbf{P}_\pi]_{ss'}\boldsymbol{\phi}(s')\right)^\top \\
&= \mathbf{Z}^\top(\mathbf{I} - \gamma\mathbf{P}_\pi)\boldsymbol{\Phi},
\end{aligned}$$

where $\mathbf{Z} \doteq [\mathbf{z}(s_1), \cdots, \mathbf{z}(s_{|\mathcal{S}|})]^\top \in \mathbb{R}^{|\mathcal{S}| \times d}$, and $\mathbf{z}(s) \in \mathbb{R}^d$ is defined by

$$
\begin{aligned}
\mathbf{z}(s) &\doteq d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu \left[\gamma\lambda \mathbf{z}_{t-1} + M_t \boldsymbol{\phi}_t | S_t = s\right] \\
&= \underbrace{d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu \left[M_t | S_t = s\right]}_{m(s)} \boldsymbol{\phi}(s) + \gamma\lambda d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu \left[\mathbf{z}_{t-1} | S_t = s\right] \\
&= m(s)\boldsymbol{\phi}(s) + \gamma\lambda d_\mu(s) \sum_{\bar{s},\bar{a}} \lim_{t\to\infty} \mathbb{P}_\mu(S_{t-1} = \bar{s}, A_{t-1} = \bar{a} | S_t = s) \mathbb{E}_\mu \left[\mathbf{z}_{t-1} | S_{t-1} = \bar{s}, A_{t-1} = \bar{a}\right] \\
&= m(s)\boldsymbol{\phi}(s) + \gamma\lambda d_\mu(s) \sum_{\bar{s},\bar{a}} \frac{d_\mu(\bar{s})\mu(\bar{a}|\bar{s})p(s|\bar{s},\bar{a})}{d_\mu(s)} \lim_{t\to\infty} \mathbb{E}_\mu \left[\mathbf{z}_{t-1} | S_{t-1} = \bar{s}, A_{t-1} = \bar{a}\right] \\
&= m(s)\boldsymbol{\phi}(s) + \gamma\lambda \sum_{\bar{s},\bar{a}} d_\mu(\bar{s})\mu(\bar{a}|\bar{s})p(s|\bar{s},\bar{a}) \frac{\pi(\bar{a}|\bar{s})}{\mu(\bar{a}|\bar{s})} \lim_{t\to\infty} \mathbb{E}_\mu \left[\gamma\lambda \mathbf{z}_{t-2} + M_{t-1}\boldsymbol{\phi}_{t-1} | S_{t-1} = \bar{s}\right] \\
&= m(s)\boldsymbol{\phi}(s) + \gamma\lambda \sum_{\bar{s}} \left(\sum_{\bar{a}} \pi(\bar{a}|\bar{s}) p(s|\bar{s},\bar{a}) \right) \mathbf{z}(\bar{s}) \\
&= m(s)\boldsymbol{\phi}(s) + \gamma\lambda \sum_{\bar{s}} [\mathbf{P}_\pi]_{\bar{s}s} \mathbf{z}(\bar{s}).
\end{aligned}
$$

In matrix form, we have

$$
\begin{aligned}
\mathbf{Z}^\top &= \boldsymbol{\Phi}^\top \mathbf{D}_\mathbf{m} + \mathbf{Z}^\top (\gamma\lambda \mathbf{P}_\pi) \\
&= \boldsymbol{\Phi}^\top \mathbf{D}_\mathbf{m} + \boldsymbol{\Phi}^\top \mathbf{D}_\mathbf{m} (\gamma\lambda \mathbf{P}_\pi) + \mathbf{Z}^\top (\gamma\lambda \mathbf{P}_\pi)^2 \\
&= \boldsymbol{\Phi}^\top \mathbf{D}_\mathbf{m} + \boldsymbol{\Phi}^\top \mathbf{D}_\mathbf{m} (\gamma\lambda \mathbf{P}_\pi) + \boldsymbol{\Phi}^\top \mathbf{D}_\mathbf{m} (\gamma\lambda \mathbf{P}_\pi)^2 + \cdots \\
&= \boldsymbol{\Phi}^\top \mathbf{D}_\mathbf{m} (\mathbf{I} - \gamma\lambda \mathbf{P}_\pi)^{-1},
\end{aligned}
$$

where $\mathbf{D}_\mathbf{m} \doteq diag(\mathbf{m}) \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$, $\mathbf{m} = [m(s_1), \cdots, m(s_{|\mathcal{S}|})]^\top \in \mathbb{R}^{|\mathcal{S}|}$, and $m(s) \in \mathbb{R}$ is defined as follows:

$$
m(s) \doteq d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu[M_t | S_t = s], \text{ for any } s \in \mathcal{S},
$$

which exists due to Lemma A.2. Further, from Lemma A.2, we have that

$$
\begin{aligned}
m(s) &= d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu[M_t | S_t = s] \\
&= d_\mu(s) \frac{d_\pi(s)}{d_\mu(s)} \\
&= d_\pi(s).
\end{aligned} \tag{A.10}
$$

In vector form, we have $\mathbf{m} = \mathbf{d}_\pi$.

Plugging $\mathbf{m} = \mathbf{d}_\pi$ and $\mathbf{Z}^\top = \boldsymbol{\Phi}^\top \mathbf{D}_\mathbf{m} (\mathbf{I} - \gamma\lambda \mathbf{P}_\pi)^{-1}$ back to the $\mathbf{A}$ matrix, we have

$$
\mathbf{A} = \boldsymbol{\Phi}^\top \mathbf{D}_\pi (\mathbf{I} - \lambda\gamma \mathbf{P}_\pi)^{-1} (\mathbf{I} - \gamma \mathbf{P}_\pi) \boldsymbol{\Phi},
$$

which is exactly the $\mathbf{A}$ matrix of On-policy TD(λ) and known to be stable (Tsitsiklis and Van Roy, 1996). Thus, LC-ETD(λ, β, ν) and its expected update are also stable by our definition.

Similarly, we can infer that

$$
\mathbf{b} = \lim_{t\to\infty} \mathbb{E}_\mu[\mathbf{b}_t] = \boldsymbol{\Phi}^\top \mathbf{D}_\pi (\mathbf{I} - \lambda\gamma \mathbf{P}_\pi)^{-1} \mathbf{r}_\pi.
$$

Note that this $\mathbf{b}$ vector is also the same as On-policy TD(λ). Thus, LC-ETD(λ, β, ν) has the same expected update as On-policy TD(λ). As a result, LC-ETD(λ, β, ν) is consistent. □

B UPDATE RULES

This section include the update rules for the algorithms mentioned in the paper.

Off-policy TD(λ):

$$\begin{aligned}
\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \alpha\delta_t\mathbf{z}_t, \\
\delta_t &= R_{t+1} + \gamma\boldsymbol{\phi}_{t+1}^\top\boldsymbol{\theta}_t - \boldsymbol{\phi}_t^\top\boldsymbol{\theta}_t, \\
\mathbf{z}_t &= \rho_t(\gamma\lambda\mathbf{z}_{t-1} + \boldsymbol{\phi}_t), \text{with } \mathbf{z}_{-1} = \mathbf{0}.
\end{aligned}$$

Full-IS-TD(λ):

$$\begin{aligned}
\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \alpha\delta_t\mathbf{z}_t, \\
\delta_t &= R_{t+1} + \gamma\boldsymbol{\phi}_{t+1}^\top\boldsymbol{\theta}_t - \boldsymbol{\phi}_t^\top\boldsymbol{\theta}_t, \\
\mathbf{z}_t &= \rho_t(\gamma\lambda\mathbf{z}_{t-1} + F_t\boldsymbol{\phi}_t), \text{with } \mathbf{z}_{-1} = \mathbf{0}, \\
F_t &= \rho_{t-1}F_{t-1}, \text{with } F_0 = 1.
\end{aligned}$$

ETD(λ):

$$\begin{aligned}
\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \alpha\delta_t\mathbf{z}_t, \\
\delta_t &= R_{t+1} + \gamma\boldsymbol{\phi}_{t+1}^\top\boldsymbol{\theta}_t - \boldsymbol{\phi}_t^\top\boldsymbol{\theta}_t, \\
\mathbf{z}_t &= \rho_t(\gamma\lambda\mathbf{z}_{t-1} + M_t\boldsymbol{\phi}_t), \text{with } \mathbf{z}_{-1} = \mathbf{0}, \\
M_t &= (1-\lambda)F_t + \lambda, \\
F_t &= \gamma\rho_{t-1}F_{t-1} + 1, \text{with } F_0 = 1.
\end{aligned}$$

ETD(λ, β):

$$\begin{aligned}
\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \alpha\delta_t\mathbf{z}_t, \\
\delta_t &= R_{t+1} + \gamma\boldsymbol{\phi}_{t+1}^\top\boldsymbol{\theta}_t - \boldsymbol{\phi}_t^\top\boldsymbol{\theta}_t, \\
\mathbf{z}_t &= \rho_t(\gamma\lambda\mathbf{z}_{t-1} + M_t\boldsymbol{\phi}_t), \text{with } \mathbf{z}_{-1} = \mathbf{0}, \\
M_t &= (1-\lambda)F_t + \lambda, \\
F_t &= \beta\rho_{t-1}F_{t-1} + 1, \text{with } F_0 = 1.
\end{aligned}$$

Scaled ETD(λ):

$$\begin{aligned}
\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \alpha\delta_t\mathbf{z}_t, \\
\delta_t &= R_{t+1} + \gamma\boldsymbol{\phi}_{t+1}^\top\boldsymbol{\theta}_t - \boldsymbol{\phi}_t^\top\boldsymbol{\theta}_t, \\
\mathbf{z}_t &= \rho_t(\gamma\lambda\mathbf{z}_{t-1} + M_t\boldsymbol{\phi}_t), \text{with } \mathbf{z}_{-1} = \mathbf{0}, \\
M_t &= (1-\lambda)F_t + \lambda(1-\gamma), \\
F_t &= \gamma\rho_{t-1}F_{t-1} + (1-\gamma), \text{with } F_0 = 1.
\end{aligned}$$

Scaled ETD(λ, β):

$$\begin{aligned}
\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \alpha\delta_t\mathbf{z}_t, \\
\delta_t &= R_{t+1} + \gamma\boldsymbol{\phi}_{t+1}^\top\boldsymbol{\theta}_t - \boldsymbol{\phi}_t^\top\boldsymbol{\theta}_t, \\
\mathbf{z}_t &= \rho_t(\gamma\lambda\mathbf{z}_{t-1} + M_t\boldsymbol{\phi}_t), \text{with } \mathbf{z}_{-1} = \mathbf{0}, \\
M_t &= (1-\lambda)F_t + \lambda(1-\beta), \\
F_t &= \beta\rho_{t-1}F_{t-1} + (1-\beta), \text{with } F_0 = 1.
\end{aligned}$$

AETD(λ):

$$
\begin{aligned}
\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \alpha\delta_t \mathbf{z}_t, \\
\delta_t &= R_{t+1} + \gamma \boldsymbol{\phi}_{t+1}^\top \boldsymbol{\theta}_t - \boldsymbol{\phi}_t^\top \boldsymbol{\theta}_t, \\
\mathbf{z}_t &= \rho_t(\gamma\lambda \mathbf{z}_{t-1} + M_t \boldsymbol{\phi}_t), \text{with } \mathbf{z}_{-1} = \mathbf{0}, \\
M_t &= (1-\lambda g(t))F_t + \lambda g(t), \\
F_t &= (1-g(t))\rho_{t-1}F_{t-1} + g(t), \text{with } F_0 = 1, \\
g(t) &= (t+1)^{-1}.
\end{aligned}
$$

LC-ETD(λ, β, ν):

$$
\begin{aligned}
\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \alpha\delta_t \mathbf{z}_t, \\
\delta_t &= R_{t+1} + \gamma \boldsymbol{\phi}_{t+1}^\top \boldsymbol{\theta}_t - \boldsymbol{\phi}_t^\top \boldsymbol{\theta}_t, \\
\mathbf{z}_t &= \rho_t(\gamma\lambda \mathbf{z}_{t-1} + M_t \boldsymbol{\phi}_t), \text{with } \mathbf{z}_{-1} = \mathbf{0}, \\
M_t &= (1-\lambda h(t))F_t + \lambda g(t), \\
F_t &= (1-g(t))\rho_{t-1}F_{t-1} + g(t), \text{with } F_0 = 1, \\
h(t) &= (1-\beta)^\nu (t+1)^{-\nu}, \\
g(t) &= (1-\beta)(t+1)^{-\nu}.
\end{aligned}
$$

LC-ETD1(λ, β):

$$
\begin{aligned}
\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \alpha\delta_t \mathbf{z}_t, \\
\delta_t &= R_{t+1} + \gamma \boldsymbol{\phi}_{t+1}^\top \boldsymbol{\theta}_t - \boldsymbol{\phi}_t^\top \boldsymbol{\theta}_t, \\
\mathbf{z}_t &= \rho_t(\gamma\lambda \mathbf{z}_{t-1} + M_t \boldsymbol{\phi}_t), \text{with } \mathbf{z}_{-1} = \mathbf{0}, \\
M_t &= (1-\lambda h(t))F_t + \lambda g(t), \\
F_t &= (1-g(t))\rho_{t-1}F_{t-1} + g(t), \text{with } F_0 = 1, \\
h(t) &= (1-\beta)^\beta (t+1)^{-\beta}, \\
g(t) &= (1-\beta)(t+1)^{-\beta}.
\end{aligned}
$$

LC-ETD2(λ, ν):

$$
\begin{aligned}
\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \alpha\delta_t \mathbf{z}_t, \\
\delta_t &= R_{t+1} + \gamma \boldsymbol{\phi}_{t+1}^\top \boldsymbol{\theta}_t - \boldsymbol{\phi}_t^\top \boldsymbol{\theta}_t, \\
\mathbf{z}_t &= \rho_t(\gamma\lambda \mathbf{z}_{t-1} + M_t \boldsymbol{\phi}_t), \text{with } \mathbf{z}_{-1} = \mathbf{0}, \\
M_t &= (1-\lambda g(t))F_t + \lambda g(t), \\
F_t &= (1-g(t))\rho_{t-1}F_{t-1} + g(t), \text{with } F_0 = 1, \\
g(t) &= (t+1)^{-\nu}.
\end{aligned}
$$

LC-ETD3(λ, β):

$$
\begin{aligned}
\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \alpha\delta_t \mathbf{z}_t, \\
\delta_t &= R_{t+1} + \gamma \boldsymbol{\phi}_{t+1}^\top \boldsymbol{\theta}_t - \boldsymbol{\phi}_t^\top \boldsymbol{\theta}_t, \\
\mathbf{z}_t &= \rho_t(\gamma\lambda \mathbf{z}_{t-1} + M_t \boldsymbol{\phi}_t), \text{with } \mathbf{z}_{-1} = \mathbf{0}, \\
M_t &= (1-\lambda g(t))F_t + \lambda g(t), \\
F_t &= (1-g(t))\rho_{t-1}F_{t-1} + g(t), \text{with } F_0 = 1, \\
g(t) &= (1-\beta)(t+1)^{-1}.
\end{aligned}
$$

C ADDITIONAL RESULTS AND EXPERIMENTAL DETAILS FOR ONE-STEP BOOTSTRAPPING

In this section, we provide additional results and experimental details to supplement the results for the one-step case in the main text. Same as Section 4, we omit the λ argument from all algorithms for notational convenience. Our Python implementations of the algorithms and environments are publicly available for future research.[1]

STABILITY OF LC-ETD(β, ν)

We use Baird's (1995) counterexample to validate the stability of LC-ETD(β, ν). Baird's counterexample is a seven-state, two-action MDP with linear features (see Figure C.1), which can illustrate the instability of Off-policy TD(λ) and other algorithms (Sutton and Barto, 2018; Jiang et al., 2022). In the one-step case, Off-policy TD(0) diverges in this example for any positive step size as long as $\gamma \in [(\sqrt{5}-1)/2, 1]$. Here, we choose $\gamma = 0.97$. Since the target policy's stationary distribution concentrates on the bottom state, the $\overline{\text{RMSVE}}$ error defined previously can only capture the errors of θ_7 and θ_8. To also take into account the errors of other dimensions of the parameter vector, we adopt the following root mean square value error as our metric: $\|\hat{\mathbf{v}}_{\boldsymbol{\theta}} - \mathbf{v}_\pi\|_{\mathbf{u}}$, where $\mathbf{u} \doteq [1/|\mathcal{S}|, \cdots, 1/|\mathcal{S}|]^\top \in \mathbb{R}^{|\mathcal{S}|}$ is a uniform distribution. We run each algorithm for 100,000 steps with the 19 step sizes mentioned in Section 4 and present the results in Figure C.2. The results are averaged over 100 independent runs, and the shaded region near each learning curve represents the standard error.

From the leftmost plot of Figure C.2, we can see that while the only existing consistent algorithm, Full-IS-TD (the green dashed line), does not learn at all as in the Two-state and Rooms tasks, LC-ETD1(β) with $\beta \in [0.2, 0.8]$ finds solutions with much lower errors. A similar observation can be found in LC-ETD2(ν). It is important to note that the importance-sampling ratio can be zero in this counterexample. This can occur with a probability of $6/7$ at any state when the agent chooses the up action. Consequently, the full IS-ratio product will quickly become zero after some time steps, and the same goes for most of the incomplete IS-ratio products. As a result, LC-ETD3(β) cannot learn because its followon trace quickly decays to an extremely small value. For Off-policy TD (the red dotted line), it diverges gradually even with very small step sizes (the smallest step size is 2^{-18}). The same goes for ETD(β) with $\beta \in [0.0, 0.4]$ (the rightmost plot), validating ETD(β)'s instability with small β. In summary, the results in Baird's counterexample highlight the stability of LC-ETD instances and illustrate the instability of ETD(β) with small β. In addition, they also show the limitation of LC-ETD(β, ν) that it cannot learn effectively with large ν when importance-sampling ratios are often zero.

EXPERIMENTAL DETAILS ON THE ROOMS TASK

Our continuing Rooms task is extended from the episodic Rooms task proposed by Ghiassian and Sutton (2021). It is based on the Four Rooms environment (Sutton et al., 1999), which can be partitioned into four parts that are connected by hallways (see Figure C.3). The Four Rooms environment has 104 states, including four hallway states. The four actions in this environment will move the agent by 1 state towards the corresponding direction. If an action causes the agent to leave the boundary, the agent will stay in the current state. The task consists of four sub-tasks. Each sub-task will assign a reward of 1 to the agent if it arrives or stays at the corresponding hallway state. However, the agent cannot stay in a hallway state

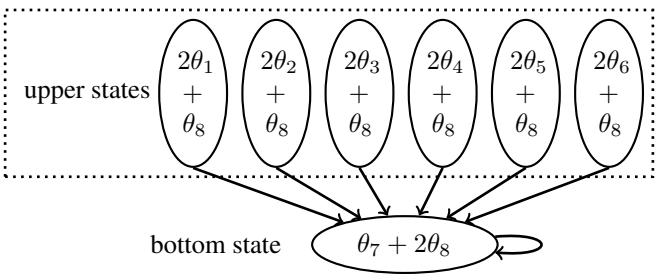


Figure C.1: Baird's counterexample. Each state has two actions. The up action will take the agent to one of the six upper states with equal probability, while the down action will take the agent to the bottom state. The target policy will choose the down action with probability 1 at any state (illustrated as the solid lines), while the probability for the behavior policy is $1/7$.

[1] See https://github.com/hejm37/LC-ETD.

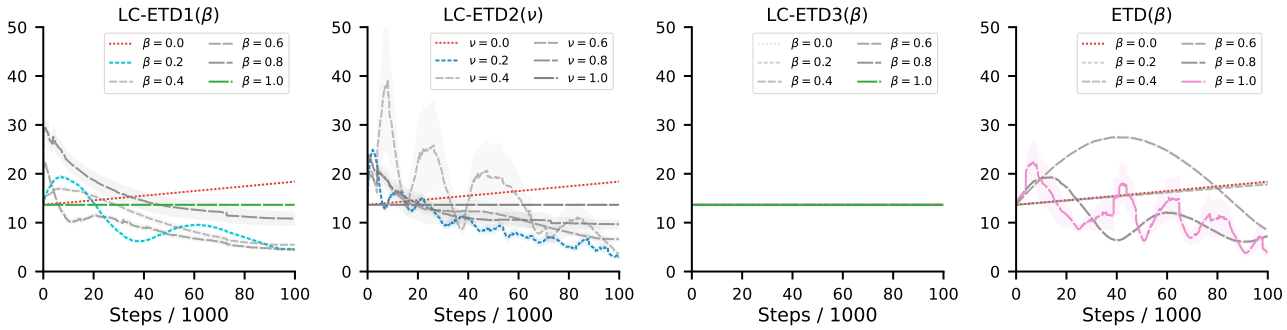


Figure C.2: Results on Baird's counterexample. The y-axis shows $\|\hat{\mathbf{v}}_{\boldsymbol{\theta}} - \mathbf{v}_\pi\|_{\mathbf{u}}$ (see text for details).

permanently as there is noise in the interactions. At each time step, there is a probability of 50% that the agent's action will be treated as one of the other three actions with equal probability. The agent needs to learn the value functions for the four target policies while following a uniform random behavior policy. The four target policies will try to go to the four hallway states. Specifically, each target policy will choose the optimal action to a corresponding hallway state with probability $1 - \epsilon$ and a random action with probability ϵ. We set ϵ to 0.1 in our experiments. The discount factor γ is 0.9. Note that it is hard to calculate the fixed points analytically in this task. Thus, we applied On-policy TD with tabular features on a trajectory following the target policy for $2{,}000{,}000$ steps for each target policy and used the final value function as the ground truth v_π. Similarly, the on-policy distributions are calculated following each target policy for $2{,}000{,}000$ steps.

SUPPLEMENTARY RESULTS ON THE ROOMS TASK

To provide a comprehensive performance profile of different one-step algorithms in the Rooms task, we present the mean results averaged over all runs in Figure C.4. From Figure C.4(a), we can see that ETD, ETD(β), LC-ETD1(β), and LC-ETD2(ν) are the top-tier algorithms in this case. Among them, ETD, LC-ETD1(β), and LC-ETD2(ν) perform less stable due to the high variance of this task. Besides, LC-ETD3(β) suffers more from the variance issue and cannot learn efficiently, but still, it learns much faster and finds much better solutions than Off-policy TD. For Full-IS-TD and Off-policy TD, their performances are not much different than the IQM results presented in Figure 4(a): The former cannot learn despite being the only existing consistent algorithm, while the latter converges to a solution with a significant bias. Finally, from Figure C.4(b), we can see that LC-ETD1(β) and LC-ETD2(ν) are still less sensitive to the decaying parameter compared to ETD(β).

SUPPLEMENTARY RESULTS OF THE BIAS-VARIANCE TRADE-OFF ANALYSIS

In this section, we explain the design choices and provide extra results for the bias-variance trade-off analysis. We first explain why we choose to study the bias and variance of trajectories of length only 30. For explanation purposes, assume that we want to analyze the bias and variance of the 2-step full IS-ratio product $F_1 = \rho_0 \rho_1$ in the Two-state task, where the target policy π will go to the left state from any state with a probability of 0.1, while the probability for the behavior policy is 0.9. Since both the target and the behavior policies are state-independent, the IS ratio ρ_t at any time step t could take a value of $1/9$ with a probability of 0.9 while choosing to go to the left state or a value of 9 with a probability of 0.1. To

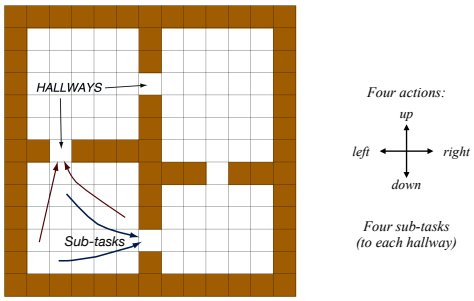
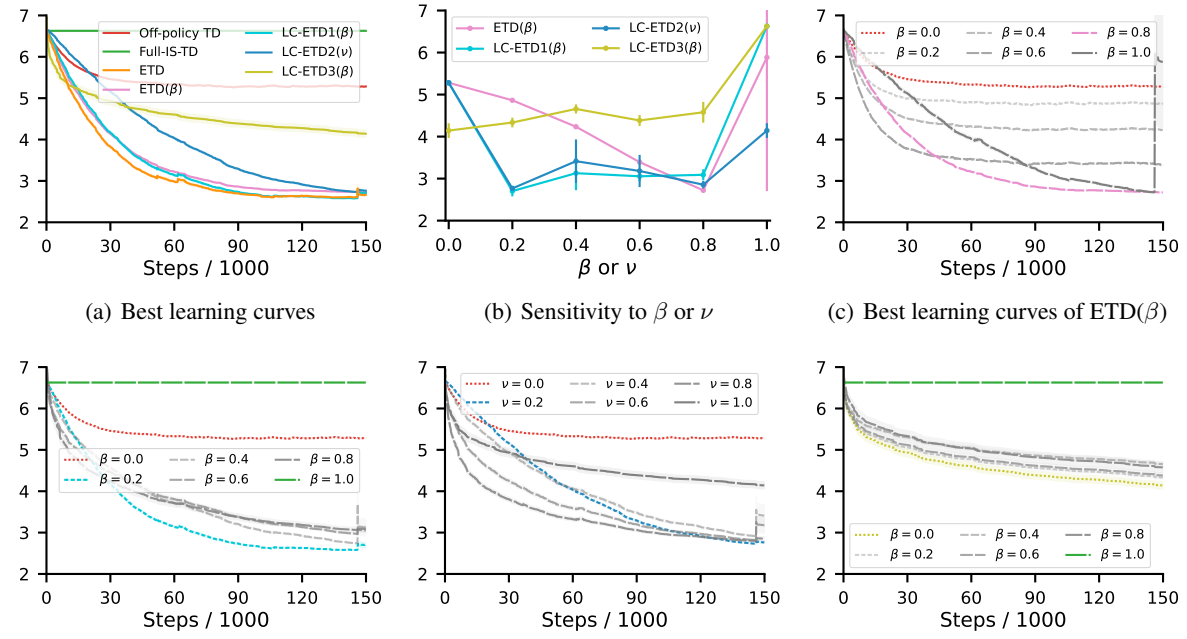


Figure C.3: The Rooms task. Modified from Sutton et al. (1999).



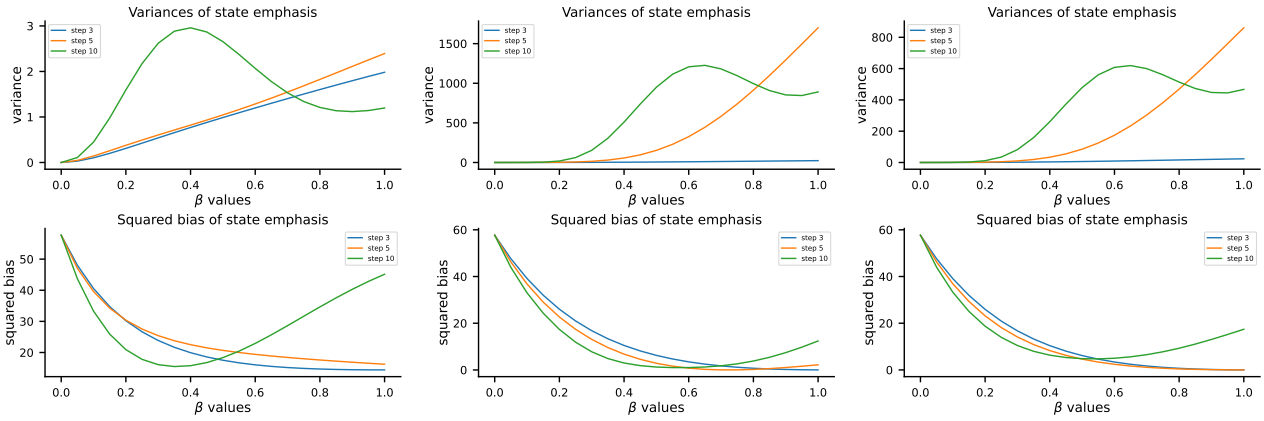
(a) Best learning curves (b) Sensitivity to β or ν (c) Best learning curves of ETD(β)

(d) Best learning curves of LC-ETD1(β) (e) Best learning curves of LC-ETD2(ν) (f) Best learning curves of LC-ETD3(β)

Figure C.4: Results averaged over all runs on the Rooms task. The y-axis shows $\overline{\text{RMSVE}}$.

obtain an accurate estimate of the mean of ρ_0 with a high probability, we will need way more than 10 seeds. To obtain an accurate estimate of the mean of F_1, we will need way more than 100 seeds. Otherwise, we can only obtain an estimation with a large bias. Thus, to obtain an accurate bias-variance analysis of different algorithms' F_t, we run experiments on short trajectories of length only 30 but with 100,000 seeds.

Next, we provide some additional experiment results to support the above discussion and reveal more insights. Figure C.5 plots the estimated bias and variance of LC-ETD1(β)'s F_t with different numbers of seeds. We can see that with 25 seeds, we can estimate the bias and variance of F_3 well but not of F_5 and F_{10}; With 5,500 seeds, the estimations of F_5's and F_{10}'s biases and variances are improved but still biased; Finally, with 12,000 seeds, we can estimate the bias and variance of F_5 well, but that of F_{10} are still biased. Now, focusing on the results of F_3 and F_5 in Figure C.5(c), we can see that as the decay parameter β increases, the bias would decrease, and the variance would increase. In addition, as the time step increases, the bias decreases, and the variance increases for any fixed β, which implies the consistency of LC-ETD1(β).

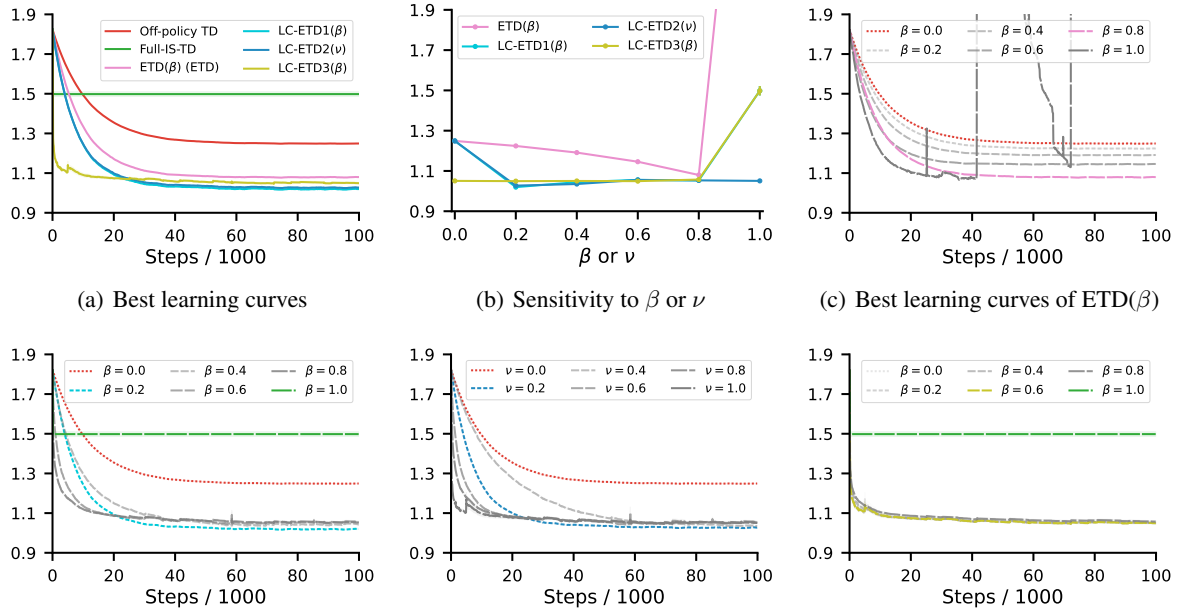


(a) 25 seeds (b) 5,500 seeds (c) 12,000 seeds

Figure C.5: The bias and variance of LC-ETD1(β)'s F_t when β varies. Label step n represents results for F_n.

D RESULTS FOR MULTI-STEP BOOTSTRAPPING

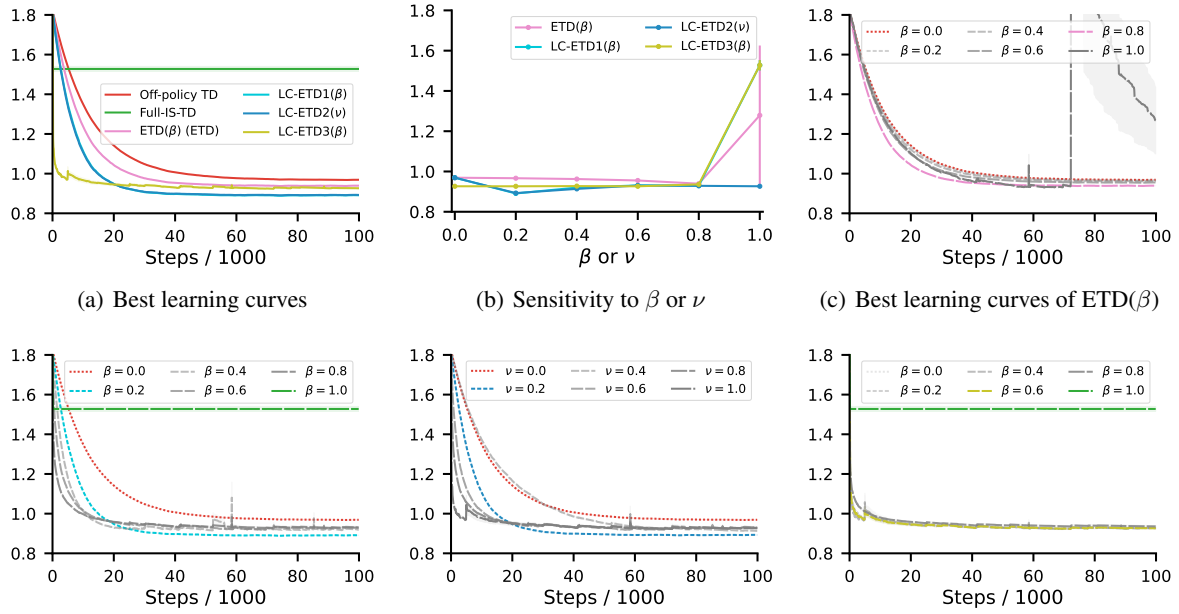
In this section, we present results for different algorithms with multi-step bootstrapping. Specifically, we studied two values of λ: $\{0.5, 0.9\}$, which correspond to different levels of bootstrapping. Figures D.1 and D.2 show the results of different algorithms with multi-step bootstrapping on the Two-state task. The conclusion is similar to the one-step case presented in Section 4 except that the biases of Off-policy TD (λ) and ETD(λ, β) reduce significantly as λ increases.



(a) Best learning curves (b) Sensitivity to β or ν (c) Best learning curves of ETD(β)

(d) Best learning curves of LC-ETD1(β) (e) Best learning curves of LC-ETD2(ν) (f) Best learning curves of LC-ETD3(β)

Figure D.1: Results on the Two-state task when $\lambda = 0.5$. The y-axis shows $\overline{\text{RMSVE}}$. The λ argument is omitted in the plots.

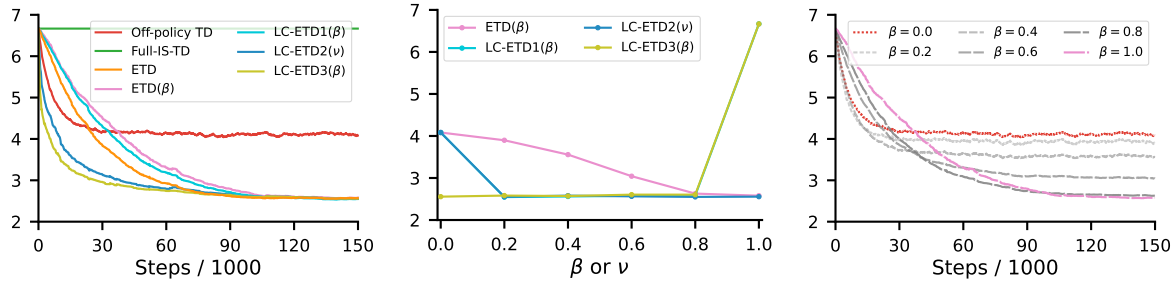


(a) Best learning curves (b) Sensitivity to β or ν (c) Best learning curves of ETD(β)

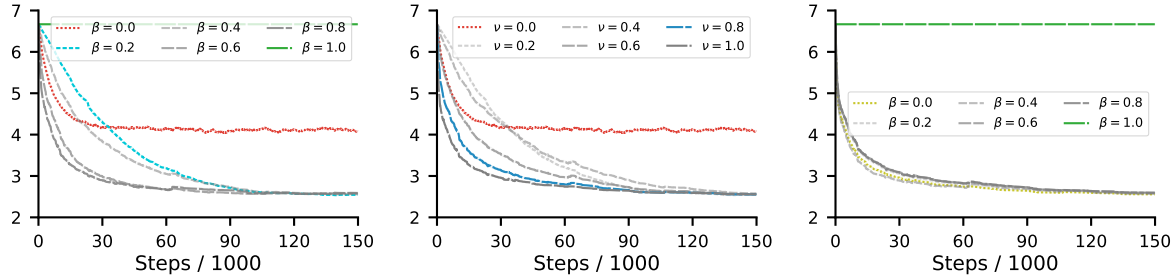
(d) Best learning curves of LC-ETD1(β) (e) Best learning curves of LC-ETD2(ν) (f) Best learning curves of LC-ETD3(β)

Figure D.2: Results on the Two-state task when $\lambda = 0.9$. The y-axis shows $\overline{\text{RMSVE}}$. The λ argument is omitted in the plots.

Figures D.3 and D.4 show the results of different algorithms with multi-step bootstrapping on the Rooms task. The conclusion is similar to the one-step case presented in Section 4 except that the biases of Off-policy TD (λ) and ETD(λ, β) reduce significantly as λ increases.

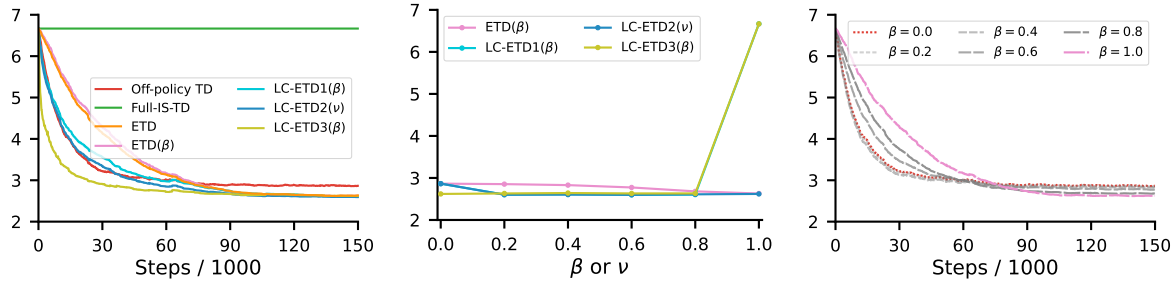


(a) Best learning curves (b) Sensitivity to β or ν (c) Best learning curves of ETD(β)

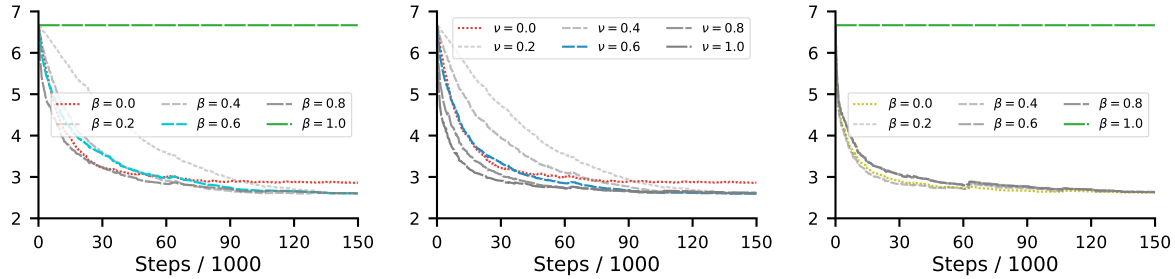


(d) Best learning curves of LC-ETD1(β) (e) Best learning curves of LC-ETD2(ν) (f) Best learning curves of LC-ETD3(β)

Figure D.3: Results on the Rooms task when $\lambda = 0.5$. The y-axis shows $\overline{\text{RMSVE}}$. The λ argument is omitted in the plots.



(a) Best learning curves (b) Sensitivity to β or ν (c) Best learning curves of ETD(β)



(d) Best learning curves of LC-ETD1(β) (e) Best learning curves of LC-ETD2(ν) (f) Best learning curves of LC-ETD3(β)

Figure D.4: Results on the Rooms task when $\lambda = 0.9$. The y-axis shows $\overline{\text{RMSVE}}$. The λ argument is omitted in the plots.

E STEP SIZE SENSITIVITY

In this section, we provide step-size sensitivity analysis on the Two-state and Rooms tasks. We aggregate the results in Figure E.1 for convenient comparisons across different dimensions. We will discuss in order the following aspects.

- The effect of an algorithm's decay parameter (β or ν) on its step-size sensitivity.
- The comparison of the step-size sensitivity of different algorithms on a single task.
- The effect of an algorithm's bootstrapping parameter (λ) on its step-size sensitivity.
- The comparison of the step-size sensitivity of different algorithms across different tasks.

Firstly, from the top-left corner of Figure E.1, we can see how different values of β affect the step-size sensitivity of LC-ETD1(0, β). Specifically, on the left extreme ($\beta = 0$), LC-ETD1(0, β) becomes Off-policy TD(0), which is the least sensitive algorithm but converges to solutions with high errors. On the right extreme ($\beta = 1$), LC-ETD1(0, β) degenerates into Full-IS-TD(0), which is the most sensitive and learns extremely slowly. While LC-ETD1(0, β) with all intermediate values of β achieves significantly lower errors, it also has an intermediate sensitivity to the step size. In summary, the sensitivity to the step size will increase as the decay parameter increase. This pattern can also be validated in other plots in the figure except for those completely flat curves that represent no sign of learning of Full-IS-TD(λ).

Next, we compare different one-step ($\lambda = 0$) algorithms' step-size sensitivity on the Two-state task from the top row of Figure E.1. It's quite obvious that ETD(0, β) is the least sensitive across different values of the decay parameter, while LC-ETD3(0, β) is at the other extreme. In addition, their best-performing step sizes for different values of the decay parameter are quite similar, which is not the case for LC-ETD1(0, β) and LC-ETD2(0, ν). Nevertheless, the latter two algorithms with a decay parameter with a value of 0.2 exhibit low sensitivity while achieving the lowest error. These observations remain valid for other rows in the figure.

Further, the leftmost plots of the top three lines provide insights into how different values of λ impact the step-size sensitivity of LC-ETD1(λ, β). Notably, as λ increases, we observe four significant findings. Firstly, LC-ETD1(λ, β) yields lower errors across different values of the decay parameter. Secondly, the method becomes increasingly sensitive to step size due to higher variance. Thirdly, the difference in error between Off-policy TD(λ) ($\beta = 0$) and LC-ETD1(λ, β) ($0 < \beta < 1$) diminishes. Finally, the sensitivity curve shifts toward smaller step sizes. These observations are consistent with those found in LC-ETD2(λ, ν) and ETD(λ, β).

Finally, we compare the sensitivity of one-step ($\lambda = 0$) algorithms to step size on two different tasks from the first and fourth rows of Figure E.1. Our observations reveal that algorithms exhibit greater sensitivity in the Rooms task, which has a higher variance than the Two-state task. This is especially notable for algorithms previously found to be less sensitive in the Two-state task. There could be two contributing factors to this observation. Firstly, the shrinkage of the suitable step size range may become smaller as the task variance increases. Alternatively, the difference could be due to how the results are summarized. We remind the reader that the results for the Two-state task were averaged over all 100 runs, while the results for the Rooms task were averaged over the middle 15 runs.

In summary, higher variance can lead to greater sensitivity to the step size. In the case of LC-ETD instances, reducing variance through a small decay parameter can improve usability. This is supported by the above analysis, which showed that a small decay parameter resulted in the lowest error while reducing sensitivity to changes in the step-size parameter. Therefore, using a small decay parameter may be an effective way to optimize the performance of LC-ETD instances.

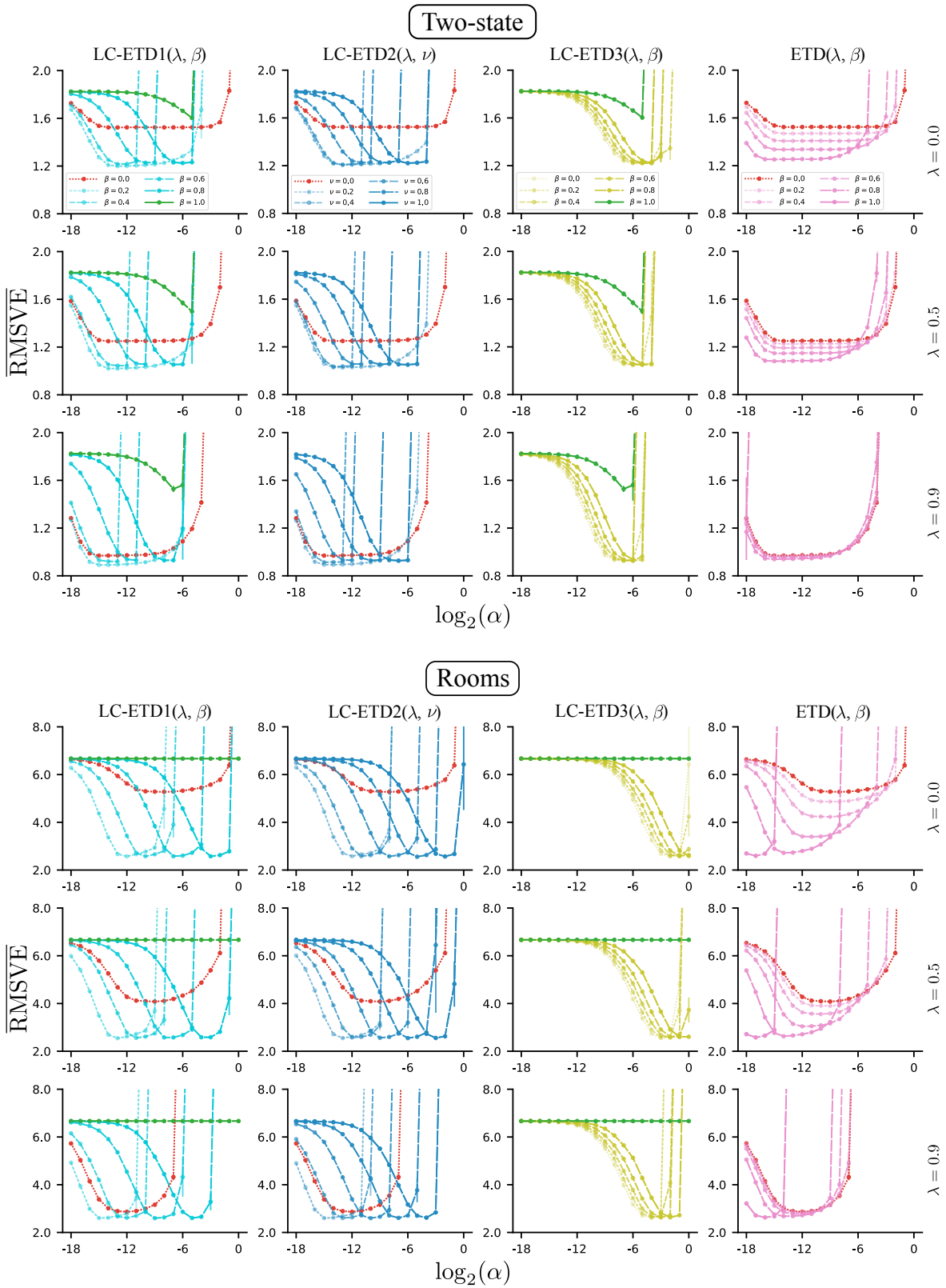


Figure E.1: Step-size sensitivity.