# Loosely Consistent Emphatic Temporal-Difference Learning

**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)

## Abstract

There has been significant interest in searching for off-policy Temporal-Difference (TD) algorithms that find the same solution that would have been obtained in the on-policy regime. An important property of such algorithms is that their expected update has the same fixed point as that of On-policy TD($\lambda$), which we call *loose consistency*. Notably, Full-IS-TD($\lambda$) is the only existing loosely consistent method under general linear function approximation but, unfortunately, has a high variance and is scarcely practical. This notorious high variance issue motivates the introduction of ETD($\lambda$), which tames down the variance but has a biased fixed point. Inspired by these two methods, we propose a new loosely consistent algorithm called *Average Emphatic TD* (AETD($\lambda$)) with a transient bias, which strikes a balance between bias and variance. Further, we unify AETD($\lambda$) with existing methods and obtain a new family of loosely consistent algorithms called *Loosely Consistent Emphatic TD* (LC-ETD($\lambda$, $\beta$, $\nu$)), which can control a smooth bias-variance trade-off by varying the speed at which the transient bias fades. Through experiments on illustrative examples, we show the effectiveness and practicality of LC-ETD($\lambda$, $\beta$, $\nu$).[1]

## 1 INTRODUCTION

Off-policy learning is a critical area in reinforcement learning (RL). Particularly, off-policy policy evaluation (OPPE), also known as off-policy prediction, is an essential component in model learning, options learning (Sutton et al., 1999), and life-long learning (Sutton et al., 2022; White et al., 2012). The goal of OPPE is to estimate the value function of a *target policy* with data collected by a different *behavior policy*. We refer to the data collected by the target policy as *on*-policy data and the data collected by the behavior policy *off*-policy data. In this paper, we consider the problem of OPPE with linear function approximation.

In online RL, where the algorithm makes incremental updates, TD learning is a ubiquitous family of algorithms, and On-policy TD($\lambda$) is an essential approach to on-policy prediction (Sutton, 1988). In OPPE, there have been substantial efforts in obtaining the *on-policy fixed point*, to which On-policy TD($\lambda$) converges with on-policy data (Precup et al., 2001; Hallak and Mannor, 2017; Gelada and Bellemare, 2019). There is a good reason for targeting the on-policy fixed point: It produces a good approximation of the target policy's value function (Tsitsiklis and Van Roy, 1996).

When the ratio between the stationary distributions of the target and behavior policies is available, we can use it to reweight the TD update, allowing for the development of algorithms that converge to the on-policy fixed point (Hallak and Mannor, 2017). However, such a ratio, known as the density ratio, is generally not accessible. One potential approach is to learn an approximation of the density ratio, which requires the ratio to be realizable by the features. When such an assumption holds, extensive studies have been conducted for both off-policy policy evaluation and optimization (Hallak and Mannor, 2017; Liu et al., 2018; Nachum et al., 2019; Zhang et al., 2020a,b; Lee et al., 2021; Zhan et al., 2022; Chen and Jiang, 2022; Huang et al., 2023).

Nevertheless, the realizability assumption on the features is quite strong and may not be feasible in practice. In this paper, we consider algorithms with theoretical guarantees that do not require such an assumption and hold under general linear function approximation. Specifically, we search for off-policy TD algorithms whose expected update has the same fixed point as the on-policy fixed point, and we say such algorithms are *loosely consistent*. An important implication of an algorithm's loose consistency is that if the algorithm converges, it is to the on-policy fixed point.

[1] The Python implementations of the experiments are available at `https://github.com/hejm37/LC-ETD`.

*Proceedings of the 39th Conference on Uncertainty in Artificial Intelligence* (UAI 2023), PMLR 216:849–859.

To our knowledge, Full Importance-Sampling TD (Full-IS-TD($\lambda$), Precup et al., 2001) is the only loosely consistent off-policy TD algorithm under general linear function approximation. To obtain the on-policy fixed point, Full-IS-TD($\lambda$) reweights the TD update with the full importance-sampling-ratio (IS-ratio) product, the multiplication of the IS ratios at every time step. However, Full-IS-TD($\lambda$) barely works in practice due to the high variance of the full IS-ratio product.

To tame down the variance of Full-IS-TD($\lambda$), Emphatic TD (ETD($\lambda$), Sutton et al., 2016) reweights the TD update with the emphatic weighting. While the emphatic weighting mitigates the variance issue, it also induces persistent bias, deviating from the on-policy TD fixed point. Further, to obtain a smooth bias-variance trade-off, Hallak et al. (2016) proposed ETD($\lambda$, $\beta$), which unifies Off-policy TD($\lambda$) and ETD($\lambda$) with a tunable parameter $\beta$. Yet, ETD($\lambda$, $\beta$) loses the stability guarantee when $\beta$ is smaller than an instance-dependent condition number that is difficult to determine.

In this paper, we first propose Average Emphatic TD (AETD($\lambda$)), a novel loosely consistent algorithm inspired by Full-IS-TD($\lambda$) and ETD($\lambda$), which strikes a better balance between bias and variance. AETD($\lambda$) renovates the idea of ETD($\lambda$), introducing a transient bias to achieve a lower variance than Full-IS-TD($\lambda$) while retaining consistency as the bias fades away over time. Then, to make AETD($\lambda$) more practical, we introduce extra parameters to control a smooth bias-variance trade-off by unifying it with existing algorithms. The resulting new family of loosely consistent algorithms called *Loosely Consistent Emphatic TD* (LC-ETD($\lambda$, $\beta$, $\nu$)) has a more general stability guarantee than ETD($\lambda$, $\beta$), the same fixed point as On-policy TD($\lambda$), and much better performance than Full-IS-TD($\lambda$). Finally, through experiments on didactic examples, we validate the stability and the benefit of loose consistency of LC-ETD($\lambda$, $\beta$, $\nu$). Experiment results on a more complex task with high variance also show LC-ETD($\lambda$, $\beta$, $\nu$)'s faster convergence to the lowest error. To our knowledge, LC-ETD($\lambda$, $\beta$, $\nu$) is the first practical, loosely consistent algorithm for off-policy TD learning under general linear function approximation.

## 2 BACKGROUND

We consider an infinite horizon Markov Decision Process (MDP), which is defined as a tuple $\langle \mathcal{S}, \mathcal{A}, p, d_0, r, \gamma \rangle$ where $\mathcal{S}$ is the finite state space, $\mathcal{A}$ is the finite action space, $p : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition function, $d_0 \in \Delta(\mathcal{S})$ is the initial state distribution, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $0 \leq \gamma < 1$ is the discount factor. Here, $\Delta(\mathcal{X})$ denotes the set of probability distributions over a finite set $\mathcal{X}$. The policy of the agent is defined as $\pi : \mathcal{S} \to \Delta(\mathcal{A})$. The discounted value function is defined as

$$v_\pi(s) \doteq \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \gamma^t r(S_t, A_t) | S_0 = s \right].$$

Particularly, we consider the setting of *online OPPE with linear function approximation*, where the agent needs to estimate the value function of a target policy $\pi$ while interacting with the environment with a behavior policy $\mu$. We assume the observation is parameterized by the feature function $\boldsymbol{\phi} : \mathcal{S} \to \mathbb{R}^d$ or equivalently the feature matrix $\boldsymbol{\Phi} \in \mathbb{R}^{|\mathcal{S}| \times d}$, where $d$ is the dimension of the feature. At each time step $t$, the agent selects action $A_t$ based on the current state $S_t$ following the behavior policy $\mu$ and observes the next state $S_{t+1}$ and reward $R_{t+1} = r(S_t, A_t)$. The *importance-sampling ratio* at time step $t$ is defined as $\rho_t \doteq \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)}$. With linear function approximation, the agent approximates the value function with $\hat{v}(s; \boldsymbol{\theta}) = \boldsymbol{\phi}(s)^\top \boldsymbol{\theta}$ or in matrix-vector form, $\hat{\mathbf{v}} \doteq \boldsymbol{\Phi}\boldsymbol{\theta}$, where $\boldsymbol{\theta} \in \mathbb{R}^d$ is a parameter vector.

We make a few common assumptions to make the problem more tractable: Firstly, Assumption 2.1 ensures the unique existence of the corresponding stationary distributions, $d_\mu \in \Delta(\mathcal{S})$ and $d_\pi \in \Delta(\mathcal{S})$. In addition, it holds that for any $s \in \mathcal{S}$, $d_\mu(s) > 0$ and $d_\pi(s) > 0$; secondly, Assumption 2.2 makes sure that $\rho_t$ is well-defined at every time step; finally, Assumption 2.3 ensures that the features are well-behaved, avoiding singularity in the analysis.

**Assumption 2.1** (Ergodicity)**.** The Markov chains induced by the behavior policy $\mu$ and the target policy $\pi$ are ergodic.

**Assumption 2.2** (Coverage)**.** For any $s \in \mathcal{S}$ and $a \in \mathcal{A}$, if $\pi(a|s) > 0$, then $\mu(a|s) > 0$.

**Assumption 2.3** (Independent Features)**.** The feature matrix $\boldsymbol{\Phi}$ has independent columns.

Let $\mathbf{I} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ denote the identity matrix, $\mathbf{P}_\pi \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ denote the on-policy transition matrix with $[\mathbf{P}_\pi]_{ss'} \doteq \sum_{a \in \mathcal{A}} \pi(a|s) p(s'|s, a)$, and $\mathbf{r}_\pi \in \mathbb{R}^{|\mathcal{S}|}$ denote the on-policy state reward function with $[\mathbf{r}_\pi]_s \doteq \sum_{a \in \mathcal{A}} \pi(a|s) r(s, a)$. Similar to identifying $\hat{v}$ as $\hat{\mathbf{v}}$, we also identify $d_\mu$ as $\mathbf{d}_\mu$ and $d_\pi$ as $\mathbf{d}_\pi$. Moreover, we define $\mathbf{D}_\mathbf{v} \doteq diag(\mathbf{v})$ for some vector $\mathbf{v}$. Specifically, we use $\mathbf{D}_\pi$ for $\mathbf{D}_{\mathbf{d}_\pi}$ and $\mathbf{D}_\mu$ for $\mathbf{D}_{\mathbf{d}_\mu}$. We use $\| \cdot \|_\mathbf{v}$ to denote the vector norm induced by $\mathbf{D}_\mathbf{v}$ for some vector $\mathbf{v}$, i.e., $\|\mathbf{x}\|_\mathbf{v} \doteq \sqrt{\mathbf{x}^\top \mathbf{D}_\mathbf{v} \mathbf{x}}$.

**Stability** We will define the stability of a stochastic algorithm with an update of the following form:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha(\mathbf{b}_t - \mathbf{A}_t \boldsymbol{\theta}_t),$$

where $\alpha > 0$ is a scalar step-size parameter, $\{\boldsymbol{\theta}_t\}_{t=0}^\infty$ is the sequence of weight vectors generated by the algorithm, and $\{(\mathbf{A}_t, \mathbf{b}_t)\}_{t=0}^\infty$ is a sequence of random matrices and vectors that depend on the problem and the algorithm. Define $\mathbf{A} \doteq \lim_{t \to \infty} \mathbb{E}_\mu[\mathbf{A}_t]$ and $\mathbf{b} \doteq \lim_{t \to \infty} \mathbb{E}_\mu[\mathbf{b}_t]$. Using $\mathbf{A}$ and $\mathbf{b}$, we can form a deterministic algorithm:

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

which we call the *expected update* of the stochastic algorithm. We use the definition of the stability of a stochastic algorithm from Sutton et al. (2016): A stochastic algorithm and its expected update are *stable* if the expected update converges to a unique fixed point under any initialization. It turns out that, the expected update is stable if and only if the eigenvalues of its $\mathbf{A}$ matrix all have positive real parts (Varga, 1999). As discussed in Sutton et al. (2016), the stability of a stochastic algorithm is essential to its convergence: If a stochastic algorithm is stable, then its parameter vector may converge with probability one with a proper step-size scheduling. Besides, if the stochastic algorithm converges, it is to the fixed point of its expected update, $\bar{\boldsymbol{\theta}} = \mathbf{A}^{-1}\mathbf{b}$. For example, under Assumptions 2.1 and 2.3, On-policy TD($\lambda$) can be shown to be stable and converge to the *on-policy fixed point*, $\bar{\boldsymbol{\theta}}_{\text{On}} = \mathbf{A}^{-1}\mathbf{b}$, where

$$\mathbf{A} = \mathbf{\Phi}^\top \mathbf{D}_\pi(\mathbf{I} - \lambda\gamma\mathbf{P}_\pi)^{-1}(\mathbf{I} - \gamma\mathbf{P}_\pi)\mathbf{\Phi} \quad \text{and}$$
$$\mathbf{b} = \mathbf{\Phi}^\top \mathbf{D}_\pi(\mathbf{I} - \lambda\gamma\mathbf{P}_\pi)^{-1}\mathbf{r}_\pi.$$

**Loose Consistency** We consider an off-policy TD algorithm to be *loosely consistent* if its expected update converges to the on-policy fixed point under any initialization.[2] By definition, loose consistency implies stability. A sufficient condition of loose consistency is that the algorithm has the same expected update (or equivalently, $\mathbf{A}$ matrix and $\mathbf{b}$ vector) as On-policy TD($\lambda$). For simplicity, we will refer to loose consistency as *consistency* and designate a loosely consistent algorithm as a *consistent algorithm* throughout the remainder of the paper.

In our pursuit of consistent off-policy TD algorithms, we next review a line of work that has made progress toward this goal by reweighting the TD update.

**Off-Policy TD($\lambda$)** Off-policy TD($\lambda$) (Precup, 2000) is the earliest effort in this line of work. In the one-step case, Off-policy TD(0) makes the following update:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha\rho_t\delta_t\boldsymbol{\phi}_t, \tag{1}$$
$$\delta_t = R_{t+1} + \gamma\boldsymbol{\phi}_{t+1}^\top\boldsymbol{\theta}_t - \boldsymbol{\phi}_t^\top\boldsymbol{\theta}_t, \tag{2}$$

where $\boldsymbol{\phi}_t \doteq \boldsymbol{\phi}(S_t)$. Compared to On-policy TD(0), Off-policy TD(0) uses the IS ratio $\rho_t$ to correct the probability of selecting action $A_t$ at time step $t$, which allows Off-policy TD(0) to converge to the true value function $v_\pi$ if the feature representation is tabular. This convergence guarantee is true for any $\lambda \in [0, 1]$ in the tabular case. However, Off-policy TD($\lambda$) can be shown to diverge in various counterexamples with general linear features (Baird, 1995; Sutton and Barto, 2018). This divergence issue is due to the distribution of Update (1) could deviate too much from the on-policy distribution, which is well explained by Sutton et al. (2016). What is worse, the *off-policy fixed point*, the fixed point that Off-policy TD($\lambda$) converges to if it does converge, could have an unbounded error in the one-step case (Kolter, 2011).

**Full-IS-TD($\lambda$)** To address the divergence of Off-policy TD($\lambda$), Precup et al. (2001) introduced the idea of correcting the distribution of Update (1) by reweighting it. The algorithm they proposed, Full-IS-TD($\lambda$), reweights the update with the *full* IS-ratio product, the product of all the IS ratios up to the current time step. Before we bring in the update of Full-IS-TD(0), we introduce a general description of one-step TD algorithms with an unspecified trace $F_t$:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha\rho_t F_t\delta_t\boldsymbol{\phi}_t, \tag{3}$$

where $\delta_t$ is the TD error defined in Eq. (2). For Off-policy TD(0), $F_t = 1$. In the case of Full-IS-TD(0), $F_t = \rho_{t-1}\rho_{t-2}\cdots\rho_0$ or $F_t = \rho_{t-1}F_{t-1}$ with $F_0 = 1$, which always corrects the distribution back to the on-policy distribution completely (Precup et al., 2001). In general, for Full-IS-TD($\lambda$), $\mathbb{E}_\mu[F_t|S_t = s]$ is equal to $\frac{\mathbb{P}_\pi(S_t=s)}{\mathbb{P}_\mu(S_t=s)}$, which converges to the density ratio $\frac{d_\pi(s)}{d_\mu(s)}$. Consequently, Full-IS-TD($\lambda$) has the same expected update as On-policy TD($\lambda$), implying its consistency. In fact, it is the only consistent method.[3] However, Full-IS-TD($\lambda$) is scarcely practical due to variance issues, which motivates the next idea in this line.

**ETD($\lambda$)** Instead of using the full IS-ratio product to weight the update, Sutton et al. (2016) proposed to use the emphatic weighting:

$$F_t = \gamma\rho_{t-1}F_{t-1} + 1, \text{with } F_0 = 1, \tag{4}$$

which is termed the *followon trace*, a geometrically weighted sum of IS-ratio products accumulated from different time steps. By introducing *incomplete* IS-ratio products into the weighting, the proposed algorithm, ETD($\lambda$), reduces the variance and remains stable. However, these incomplete IS-ratio products also introduce persistent bias into $F_t$, causing ETD($\lambda$) to converge to a biased fixed point.

# 3 TOWARDS PRACTICAL, CONSISTENT TD LEARNING

## AVERAGE FOLLOWON TRACE

As discussed in the last section, the only consistent method, Full-IS-TD($\lambda$), is not practical due to the high variance issue. On the other hand, ETD($\lambda$), an effective remedy to the variance issue, is biased and deviates from our objective

[2] Hallak and Mannor (2017) referred to a similar property as "consistency." However, to distinguish it from the usage of the word "consistency" in statistics, we redefine it as "loose consistency." Following the statistical convention, we can define strong or weak consistencies based on whether the algorithm converges almost surely or in probability.

[3] From now on, our discussion will be based exclusively on Assumption 2.3 for the features, without any additional assumptions.

of finding the on-policy fixed point. Then, *can we find a good trade-off point between Full-IS-TD($\lambda$) and ETD($\lambda$)?* Specifically, can we find a method that is consistent and has a milder variance compared to Full-IS-TD($\lambda$)? The answer is, fortunately, yes. Inspired by the idea of using incomplete IS-ratio products to reduce the variance, we propose to use the below average emphatic weighting:

$$F_t = \frac{t}{t+1}\rho_{t-1}F_{t-1} + \frac{1}{t+1}, \text{with } F_0 = 1, \qquad (5)$$

which we term the *average followon trace*. Expanding this trace reveals that it represents the mean of the IS-ratio products. By employing the mean instead of a geometrically weighted sum, we gradually reduce bias by diminishing the emphasis on the new IS-ratio product at each time step. Although the expectation of the average followon trace at time step $t$ typically differs from $\frac{\mathbb{P}_\pi(S_t=s)}{\mathbb{P}_\mu(S_t=s)}$, this discrepancy diminishes as $t$ increases, characterizing the average followon trace as displaying a fading or transient bias. Remarkably, the bias of $F_t$ completely vanishes in the limit, rendering the resulting algorithm defined by Update (3) and Eq. (5) consistent. This algorithm is referred to as *one-step Average Emphatic TD* (AETD(0)), and its consistency is presented in Theorem 3.1. The detailed proof is deferred to the appendix.

**Theorem 3.1** (Consistency of AETD(0)). *Let Assumptions 2.1-2.3 hold. If $\lim_{t\to\infty}\mathbb{E}_\mu[F_t|S_t = s]$ exists for all $s \in \mathcal{S}$, then AETD(0) has the same expected update as On-policy TD(0). As a result, AETD(0) is stable and consistent.*

Now, the idea of using a uniformly weighted sum of the IS-ratio products to reweight the TD update is not entirely new. Hallak et al. (2016) unified ETD($\lambda$) and Off-policy TD($\lambda$) by introducing a tunable decay parameter, $\beta \in [0, 1]$, in the followon trace (Eq. (4)). The resulting algorithm, ETD($\lambda$, $\beta$), uses the following emphatic weighting:

$$F_t = \beta\rho_{t-1}F_{t-1} + 1, \text{with } F_0 = 1. \qquad (6)$$

When $\beta = 0$, this trace degenerates to constant 1, and ETD($\lambda$, $\beta$) becomes Off-policy TD($\lambda$); when $\beta = \gamma$, this trace recovers Eq. (4), and ETD($\lambda$, $\beta$) becomes ETD($\lambda$); when $\beta = 1$, this trace will equally weight each IS-ratio product with weight 1. However, in their case, equally weighting the products is problematic because the expectation of $F_t$ diverges to infinity in the limit.

## A SMOOTH BIAS-VARIANCE TRADE-OFF

Despite ETD($\lambda$, $\beta$) not being a consistent algorithm for any value of $\beta$, it presents an interesting strategy to trade off bias and variance: With a small $\beta$, $F_t$ has a low variance but a large bias; with a large $\beta$, $F_t$ has a small bias but a high variance. Consequently, ETD($\lambda$, $\beta$) can trade off the bias of its fixed point and the variance it incurs by varying the value of $\beta$. Then, we wonder, *can we unify AETD(0) with other algorithms to attain a smooth bias-variance trade-off?* If possible, we also want to retain the consistency of AETD(0). To achieve this goal, we consider the unification of AETD(0) with both Off-policy TD(0) and Full-IS-TD(0), the one with the least variance but the greatest bias and the one with the least bias but the greatest variance.

We first unify AETD(0) and Full-IS-TD(0). To unify the traces that the two methods use, we introduce a tunable parameter, $\beta' \in [0, 1]$, to the average followon trace: $F_t^{(1)} = (1 - \beta'(t + 1)^{-1})\rho_{t-1}F_{t-1}^{(1)} + \beta'(t + 1)^{-1}$ with $F_0^{(1)} = 1$. Then, when $\beta' = 0$, $F_t^{(1)}$ becomes $F_t^{(1)} = \rho_{t-1}F_{t-1}^{(1)}$, which corresponds to the trace of Full-IS-TD(0); when $\beta' = 1$, $F_t^{(1)}$ becomes the average followon trace.

Similarly, we can unify AETD(0) and Off-policy TD(0) with another tunable parameter, $\nu \in [0, 1]$, and a new trace: $F_t^{(2)} = (1-(t+1)^{-\nu})\rho_{t-1}F_{t-1}^{(2)} + (t+1)^{-\nu}$ with $F_0^{(2)} = 1$. When $\nu = 0$, $F_t^{(2)}$ becomes constant 1, which corresponds to the trace of Off-policy TD(0); when $\nu = 1$, $F_t^{(2)}$ becomes the average followon trace.

We further unify $F_t^{(1)}$ and $F_t^{(2)}$, leading us to a third trace with two parameters, $\beta'$ and $\nu$: $F_t^{(3)} = (1 - \beta'(t + 1)^{-\nu})\rho_{t-1}F_{t-1}^{(3)} + \beta'(t + 1)^{-\nu}$ with $F_0^{(3)} = 1$. Additionally, we found that when $\nu = 0$, the trace becomes $F_t = (1 - \beta')\rho_{t-1}F_{t-1} + \beta'$, which is also a geometrically weighted sum of IS-ratio products as in ETD($\lambda$, $\beta$). To obtain the same decay rate in the resulting trace and the followon trace (Eq. (6)), we replace $\beta'$ with $1 - \beta$ in $F_t^{(3)}$, and name the resulting trace *general followon trace*:

$$F_t = (1 - g(t))\rho_{t-1}F_{t-1} + g(t), \text{with } F_0 = 1, \qquad (7)$$

where $g(t) \doteq (1 - \beta)(t + 1)^{-\nu}$ with $\beta \in [0, 1]$ and $\nu \in [0, 1]$. Note that when $\nu = 0$, the resulting trace becomes $F_t = \beta\rho_{t-1}F_{t-1} + (1-\beta)$, which we call the scaled followon trace. The resulting one-step algorithm is subsequently called Scaled ETD(0, $\beta$). Although the scaled followon trace has the same decay rate as the original followon trace (Eq. (6)), it is downscaled by $1 - \beta$ (see Table 1). This discrepancy, however, is not a qualitative difference because the constant factor $1 - \beta$ can be absorbed in the step-size parameter.[4] Thus, Scaled ETD(0, $\beta$) can be viewed as a slight variant of ETD(0, $\beta$).

Having settled the relationship between Scaled ETD(0, $\beta$) and ETD(0, $\beta$), we are now ready to name the algorithm that unifies AETD(0), Off-policy TD(0), Full-IS-TD(0), and Scaled ETD(0, $\beta$). We call the resulting algorithm *one-step General Emphatic TD* (GETD(0, $\beta$, $\nu$)), which is defined by Update (3) and the general followon trace (Eq. (7)).

[4] We can see from Table 1 that the coefficient of the full IS-ratio product, $\beta^t$, is not downscaled to $\beta^t(1 - \beta)$. However, this minor difference will not prevent Scaled ETD(0, $\beta$) from sharing the same theory and empirical properties as ETD(0, $\beta$).

Table 1: The coefficients of different IS-ratio products in $F_t$.

| IS-ratio Product | Off-policy TD($\lambda$) | Scaled ETD($\lambda$, $\beta$) | Full-IS-TD($\lambda$) | AETD($\lambda$) | LC-ETD($\lambda$, $\beta$, $\nu$) | ETD($\lambda$, $\beta$) |
|---|---|---|---|---|---|---|
| $1$ | $1$ | $1-\beta$ | $0$ | $1/(t+1)$ | $g(t)$ | $1$ |
| $\rho_{t-1}$ | $0$ | $\beta(1-\beta)$ | $0$ | $1/(t+1)$ | $(1-g(t))g(t-1)$ | $\beta$ |
| $\rho_{t-1}\rho_{t-2}$ | $0$ | $\beta^2(1-\beta)$ | $0$ | $1/(t+1)$ | $\Pi_{k=t-1}^{t}(1-g(k))g(t-2)$ | $\beta^2$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $\Pi_{k=2}^{t}\rho_{k-1}$ | $0$ | $\beta^{t-1}(1-\beta)$ | $0$ | $1/(t+1)$ | $\Pi_{k=2}^{t}(1-g(k))g(1)$ | $\beta^{t-1}$ |
| $\Pi_{k=1}^{t}\rho_{k-1}$ | $0$ | $\beta^t$ | $1$ | $1/(t+1)$ | $\Pi_{k=1}^{t}(1-g(k))$ | $\beta^t$ |

So far, we have only introduced the one-step form of AETD and GETD. By applying the same idea of uniform averaging and the same strategy of unification to the multi-step bootstrapping case, we can obtain their multi-step version. Here, we present the unified algorithm with multi-step bootstrapping called *General Emphatic TD* (GETD($\lambda$, $\beta$, $\nu$)), which makes the following update[5]:

$$\begin{aligned}
\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \alpha\delta_t \mathbf{z}_t, \\
\delta_t &= R_{t+1} + \gamma\boldsymbol{\phi}_{t+1}^\top\boldsymbol{\theta}_t - \boldsymbol{\phi}_t^\top\boldsymbol{\theta}_t, \\
\mathbf{z}_t &= \rho_t(\gamma\lambda\mathbf{z}_{t-1} + M_t\boldsymbol{\phi}_t), \text{with } \mathbf{z}_{-1} = \mathbf{0}, \qquad (8)\\
M_t &= (1-\lambda h(t))\, F_t + \lambda g(t), \\
F_t &= (1-g(t))\, \rho_{t-1}F_{t-1} + g(t), \text{with } F_0 = 1,
\end{aligned}$$

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

$$h(t) \doteq \left(\frac{1-\beta}{t+1}\right)^{\nu} \text{ and } g(t) \doteq \frac{1-\beta}{(t+1)^{\nu}} \qquad (9)$$

with $\beta \in [0,1]$ and $\nu \in [0,1]$. Similar to the one-step case, GETD($\lambda$, $\beta$, $\nu$) subsumes AETD($\lambda$) , Off-policy TD($\lambda$), Full-IS-TD($\lambda$), and Scaled ETD($\lambda$, $\beta$). A list of the updates of all these algorithms is included in the appendix.

## LOOSELY CONSISTENT EMPHATIC TD

In this section, we examine the product of the unification. The question here is, *while the introduced decay parameters $\beta$ and $\nu$ offer us a smooth bias-variance trade-off, is the consistency of AETD(0) retained?* Fortunately, the answer is, again, yes. We name this new class of consistent algorithms with tunable decay parameters as *Loosely Consistent Emphatic TD* (LC-ETD($\lambda$, $\beta$, $\nu$)).[6] Specifically, LC-ETD($\lambda$, $\beta$, $\nu$) is defined by Update (8) with $\beta \in [0,1)$ and $\nu \in (0,1]$, or $\beta = 1$ and $\nu \in [0,1]$.[7] We provide its pseudocode in Algorithm 1 and present its consistency in Theorem 3.2, of which the proof is deferred to the appendix.

**Algorithm 1:** LC-ETD($\lambda$, $\beta$, $\nu$) for online OPPE with linear function approximation

```
Input MDP $\langle \mathcal{S}, \mathcal{A}, p, d_0, r, \gamma \rangle$, feature function $\boldsymbol{\phi}$,
    behavior policy $\mu$, target policy $\pi$, step size
    $\alpha \in (0,1]$, bootstrapping parameter $\lambda \in [0,1]$,
    and decay parameters $\beta \in [0,1)$ and $\nu \in (0,1]$,
    or $\beta = 1$ and $\nu \in [0,1]$
Initialize value-function weights $\boldsymbol{\theta}$ arbitrarily,
    followon trace $F = 1$, and eligibility trace $\boldsymbol{z} = \mathbf{0}$
Draw $S_0$ from $d_0$
for $t = 0 : \infty$ do
    Take action $A_t \sim \mu(\cdot|S_t)$
    Observe $S_{t+1} \sim p(\cdot|S_t, A_t)$, $R_{t+1} = r(S_t, A_t)$
    $M \leftarrow (1-\lambda h(t))F + \lambda g(t)$, where
      $g(t) = (1-\beta)(t+1)^{-\nu}$ and
      $h(t) = (1-\beta)^{\nu}(t+1)^{-\nu}$
    $\boldsymbol{z} \leftarrow \rho_t(\gamma\lambda\boldsymbol{z} + M\boldsymbol{\phi}(S_t))$, where $\rho_t = \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)}$
    $F \leftarrow (1-g(t+1))\rho_t F + g(t+1)$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha[R_{t+1} + \gamma\boldsymbol{\phi}(S_{t+1})^\top\boldsymbol{\theta} - \boldsymbol{\phi}(S_t)^\top\boldsymbol{\theta}]\boldsymbol{z}$
end
```

**Theorem 3.2** (Consistency of LC-ETD($\lambda$, $\beta$, $\nu$))**.** *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($\lambda$, $\beta$, $\nu$) has the same expected update as On-policy TD($\lambda$). As a result, LC-ETD($\lambda$, $\beta$, $\nu$) is stable and consistent.*

*Remark* 3.3. LC-ETD($\lambda$, $\beta$, $\nu$) is stable for any values of $\beta \in [0,1)$ and $\nu \in (0,1]$, or $\beta = 1$ and $\nu \in [0,1]$. This is significantly stronger than ETD($\lambda$, $\beta$) (Hallak et al., 2016). In their case, ETD($\lambda$, $\beta$) is stable only with $\beta > \beta_0$ where $\beta_0 \leq \gamma$ is an instance-dependent condition number.

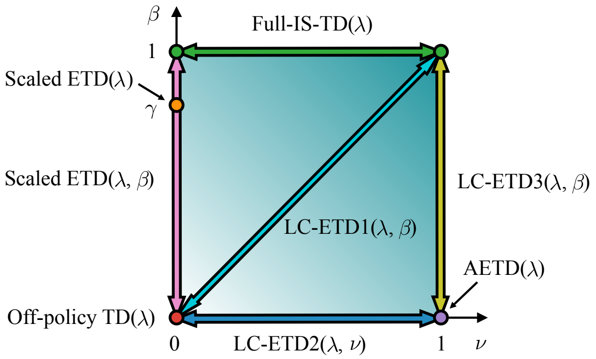
*Remark* 3.4. LC-ETD($\lambda$, $\beta$, $\nu$) is consistent for any values of $\beta \in [0,1)$ and $\nu \in (0,1]$, or $\beta = 1$ and $\nu \in [0,1]$. This is, again, significantly stronger than ETD($\lambda$, $\beta$). For any $\beta \in [0,1)$, ETD($\lambda$, $\beta$) has persistent bias. In particular, the bias will increase as the value of $\beta$ decrease. At the extreme end when $\beta = 0$, ETD($\lambda$, $\beta$) becomes Off-policy TD($\lambda$), which could have unbounded bias (Kolter, 2011).

Having settled the consistency of LC-ETD($\lambda$, $\beta$, $\nu$), we now discuss the bias-variance trade-off we obtained. Figure 1

[5] For simplicity, we have not included general state-dependent interest, discounting, and bootstrapping functions as Sutton et al. (2016). However, GETD($\lambda$, $\beta$, $\nu$) can be extended to those cases.

[6] Recall that we refer to loose consistency as consistency.

[7] We also include Full-IS-TD($\lambda$) in LC-ETD($\lambda$, $\beta$, $\nu$), since it is also consistent. LC-ETD($\lambda$, $\beta$, $\nu$) becomes Full-IS-TD($\lambda$) when $\beta = 1$ and $\nu \in [0,1]$.



$\beta$
Full-IS-TD($\lambda$)
1
Scaled ETD($\lambda$)
$\gamma$
Scaled ETD($\lambda$, $\beta$)
LC-ETD3($\lambda$, $\beta$)
LC-ETD1($\lambda$, $\beta$)
AETD($\lambda$)
Off-policy TD($\lambda$)
0
LC-ETD2($\lambda$, $\nu$)
1
$\nu$

Figure 1: The landscape of GETD($\lambda$, $\beta$, $\nu$). The square excluding the left edge and its bottom endpoint represents LC-ETD($\lambda$, $\beta$, $\nu$). The darkness of the color at each point inside the square represents the magnitude of $F_t$'s variance.

plots the landscape of GETD($\lambda$, $\beta$, $\nu$), which illustrates the relationship between LC-ETD($\lambda$, $\beta$, $\nu$) and other algorithms. Starting from AETD($\lambda$), intuitively, as $\nu$ decreases, the algorithm gets closer to Off-policy TD($\lambda$) with the variance decreased, but the bias increases; meanwhile, as $\beta$ increases, the algorithm moves towards to Full-IS-TD($\lambda$) with the bias decreased, but the variance increased. More generally, it holds for LC-ETD($\lambda$, $\beta$, $\nu$) that increasing $\beta$ or $\nu$ will reduce the bias and increase the variance, and vice versa.

To better analyze the bias-variance trade-off that $\beta$ and $\nu$ control, we study three instances of LC-ETD($\lambda$, $\beta$, $\nu$), which cover a diagonal line and two edges of LC-ETD($\lambda$, $\beta$, $\nu$) (see Figure 1). The first instance is LC-ETD1($\lambda$, $\beta$), which corresponds to a diagonal line of LC-ETD($\lambda$, $\beta$, $\nu$). In this diagonal line, the value of $\nu$ is always the same as the value of $\beta$. This line has the special property that it connects Off-policy TD($\lambda$) and Full-IS-TD($\lambda$). The update of LC-ETD1($\lambda$, $\beta$) is the same as Update (8) but with $h(t)$ and $g(t)$ specified as the following:

$$h(t) \doteq \left(\frac{1-\beta}{t+1}\right)^{\beta} \text{ and } g(t) \doteq \frac{1-\beta}{(t+1)^{\beta}}. \tag{10}$$

The second instance is LC-ETD2($\lambda$, $\nu$), the bottom edge of LC-ETD($\lambda$, $\beta$, $\nu$), which connects Off-policy TD($\lambda$) and AETD($\lambda$). Here, $\beta$ is always 0. The update of LC-ETD2($\lambda$, $\nu$) is the identical as Update (8) but with $h(t)$ and $g(t)$ set as the following:

$$h(t) \doteq (t+1)^{-\nu} \text{ and } g(t) \doteq (t+1)^{-\nu}. \tag{11}$$

The third instance is LC-ETD3($\lambda$, $\beta$), the right edge of LC-ETD($\lambda$, $\beta$, $\nu$), which links AETD($\lambda$) and Full-IS-TD($\lambda$). In this edge, $\nu$ is always 1. The update of LC-ETD3($\lambda$, $\beta$) is the same as Update (8) but with $h(t)$ and $g(t)$ specified as the following:

$$h(t) \doteq \frac{1-\beta}{t+1} \text{ and } g(t) \doteq \frac{1-\beta}{t+1}. \tag{12}$$

# 4 EXPERIMENTS

In this section, we present experiments that demonstrate the effectiveness of LC-ETD($\lambda$, $\beta$, $\nu$) in the one-step case. The results for the multi-step case exhibit a similar pattern and are provided in the appendix. Additionally, for stability analysis on Baird's (1995) counterexample, please refer to the appendix. To maintain simplicity, we omit the $\lambda$ argument from all algorithms. For instance, LC-ETD($\beta$, $\nu$) refers to LC-ETD(0, $\beta$, $\nu$). We evaluate the quality of the learned $\boldsymbol{\theta}$ using the root-mean-square-value error as our metric:

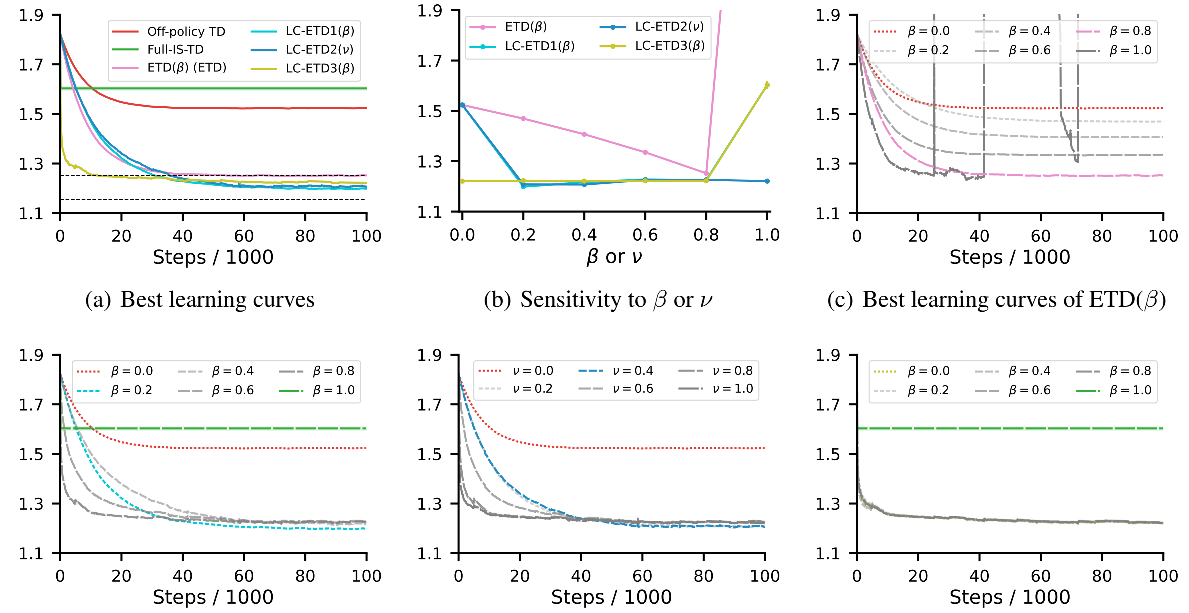
$$\overline{\text{RMSVE}}(\boldsymbol{\theta}) = \|\hat{\mathbf{v}}_{\boldsymbol{\theta}} - \mathbf{v}_{\pi}\|_{\mathbf{d}_{\pi}}.$$

For all experiments, we use constant step sizes $\alpha = 2^x$ for all algorithms where $x \in \{-18, -17, \cdots, -1, 0\}$. For *tunable algorithms* with an adjustable decay parameter (ETD($\beta$), LC-ETD1($\beta$), LC-ETD2($\nu$), and LC-ETD3($\beta$)), the decay parameter ($\beta$ or $\nu$) is chosen from $\{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. Note that ETD($\beta$), LC-ETD1($\beta$), and LC-ETD2($\nu$) with $\beta = 0.0$ or $\nu = 0.0$ are the same as Off-policy TD; LC-ETD1($\beta$) and LC-ETD3($\beta$) with $\beta = 1.0$ are the same as Full-IS-TD; ETD($\beta$) with $\beta = 1.0$ is an unsound method with a followon trace whose expectation will blow up to infinity in the limit. All results are reported with the best-performing step size, with which the final error is the smallest. We also provide the step-size sensitivity analysis in the appendix. The final error is calculated by averaging the errors in the last $1\%$ of the training steps. Compared to the area under the learning curve (AUC), the final error is favored because it is a better reflection of how the algorithm performs asymptotically.

### CONSISTENCY OF LC-ETD($\beta$, $\nu$)

**Two-State Task** To illustrate the benefit of LC-ETD($\beta$, $\nu$)'s consistency, we designed a didactic task with two states (Figure 3). In this task, the target policy $\pi$ will go to the left state from any state with a probability of 0.6, while the probability for the behavior policy $\mu$ is 0.4. The discount factor $\gamma$ is 0.8. The on-policy fixed point in this task induces an error of $\overline{\text{RMSVE}}(\bar{\theta}_{\text{On}}) \approx 1.155$, whereas the off-policy fixed point induces an error of $\overline{\text{RMSVE}}(\bar{\theta}_{\text{Off}}) \approx 1.523$. For ETD (ETD($\beta$) with $\beta = \gamma = 0.8$), its fixed point has an error of $\overline{\text{RMSVE}}(\bar{\theta}_{\text{ETD}}) \approx 1.251$. Thus, consistent algorithms have a theoretical advantage in this task because their fixed point (the on-policy fixed point) has the lowest $\overline{\text{RMSVE}}$. We run each algorithm for 100,000 steps and present the results in Figure 2, which are averaged over 100 independent runs. The shaded region near each learning curve represents the standard error. Likewise, the standard error is shown as an error bar for each point in the sensitivity plot.

From Figure 2(a), we can see that all LC-ETD instances achieve an error between $\overline{\text{RMSVE}}(\bar{\theta}_{\text{ETD}}) \approx 1.251$ and $\overline{\text{RMSVE}}(\bar{\theta}_{\text{On}}) \approx 1.155$ (the dash lines). They are the best-performing algorithms and significantly improve over the



(a) Best learning curves

(b) Sensitivity to $\beta$ or $\nu$

(c) Best learning curves of ETD($\beta$)

(d) Best learning curves of LC-ETD1($\beta$)

(e) Best learning curves of LC-ETD2($\nu$)

(f) Best learning curves of LC-ETD3($\beta$)

Figure 2: Performance of different algorithms on the Two-state task. The y-axis shows $\overline{\text{RMSVE}}$. The dash lines from top to bottom in Figure (a) show $\overline{\text{RMSVE}}(\bar{\theta}_{\text{ETD}}) \approx 1.251$ and $\overline{\text{RMSVE}}(\bar{\theta}_{\text{On}}) \approx 1.155$, respectively.

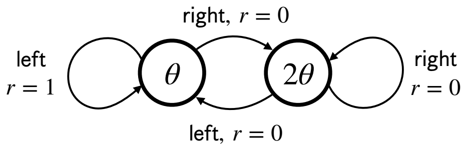


Figure 3: The Two-state task. The values of the two states are approximated by $\theta$ and $2\theta$, respectively.

only existing consistent algorithm, Full-IS-TD, which cannot learn due to the high variance issue. On the other hand, ETD (ETD($\beta$) with $\beta = 0.8$) is the second-tier algorithm in this task, achieving its theoretical optimal error of 1.251. For Off-policy TD, it also converges to its fixed point, which induces a significantly larger error of $\overline{\text{RMSVE}}(\bar{\theta}_{\text{Off}}) \approx 1.523$.

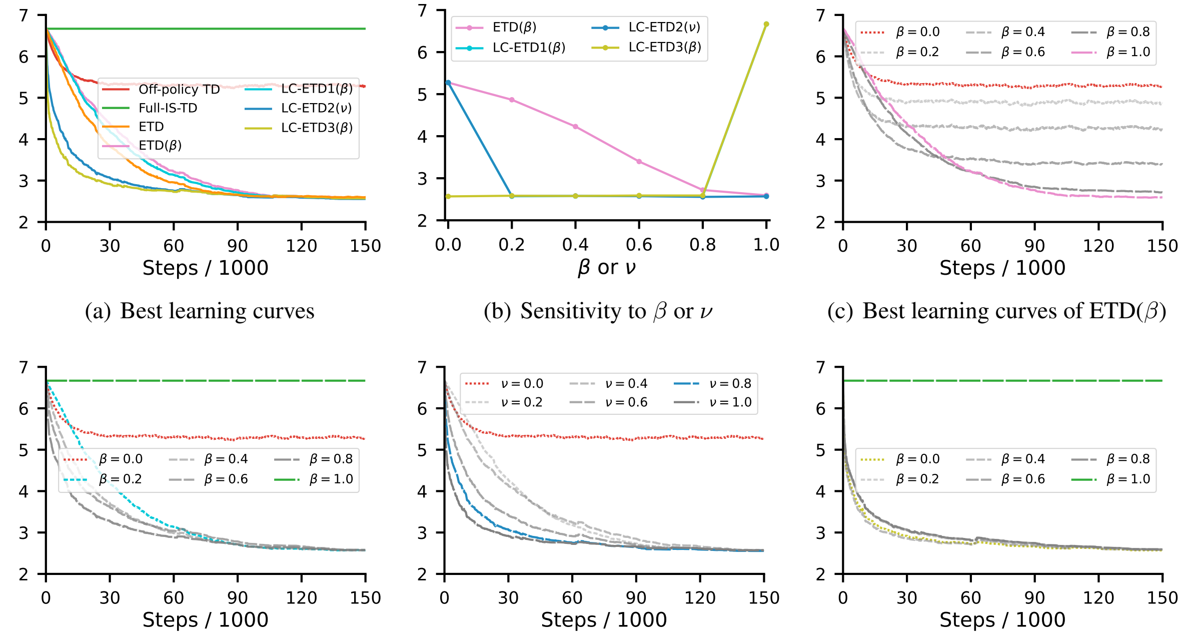
Figures 2(c)-2(f) plot the learning curves of tunable algorithms with fixed values of the decay parameter. From Figure 2(c), it is evident that ETD($\beta$) converges to solutions with large biases for most values of $\beta$. When $\beta = 1$, its error explodes after some steps, demonstrating the unsoundness of ETD($\beta$) with $\beta = 1$ in the infinite horizon case. For LC-ETD instances (Figures 2(d)-2(f)), with the decay parameter in interval $[0.2, 0.8]$, they all converge smoothly to errors at a similar level, which is lower than existing algorithms.

Figure 2(b) summarizes these results. We can conclude that all LC-ETD instances consistently enjoy lower errors than existing algorithms, which implies LC-ETD($\beta$, $\nu$)'s consistency across all its decay parameter choices.

## PRACTICALITY OF LC-ETD($\beta$, $\nu$)

**Rooms Task** To further test the performance of LC-ETD($\beta$, $\nu$) in more complex tasks with higher variance, we modified the Rooms task proposed by Ghiassian and Sutton (2021) to include continuing target policies. The discount factor $\gamma$ is kept at 0.9. Other task specifications also largely follow from Ghiassian and Sutton (2021), and the modifications can be found in the appendix. Compared to the Two-state task, the Rooms task has more states and complex feature representation. Moreover, the differences between the target policies and the behavior policy are larger, inducing much larger variance. We run each algorithm for 150,000 steps and 30 runs. To better illustrate the advantage of LC-ETD($\beta$, $\nu$), we present the results using Interquartile Mean (IQM) in Figure 4, which are more robust and statistically efficient compared to the mean or median results (Agarwal et al., 2021). The standard error is presented as a shaded region or an error bar, similar to the Two-state task. However, in this case, the standard error is based only on the middle 50% of the samples since we used IQM instead of the mean. Additionally, we provide a comprehensive performance profile by presenting the mean results (averaged over all runs) in the appendix.

Figure 4(a) shows that ETD, ETD($\beta$), and all LC-ETD instances achieve similar final errors. Among them, LC-ETD2($\nu$) and LC-ETD3($\beta$) learn the fastest. Same as in the Two-state task, Off-policy TD converges quickly to a solution with a large bias, while Full-IS-TD cannot learn.



(a) Best learning curves (b) Sensitivity to $\beta$ or $\nu$ (c) Best learning curves of ETD($\beta$)

(d) Best learning curves of LC-ETD1($\beta$) (e) Best learning curves of LC-ETD2($\nu$) (f) Best learning curves of LC-ETD3($\beta$)

Figure 4: Performance of different algorithms on the Rooms task. The y-axis shows $\overline{\text{RMSVE}}$.

Figures 4(c)-4(f) plot the learning curves of tunable algorithms with fixed values of the decay parameter. From Figure 4(c), we can see that as $\beta$ increases, the bias of the solution ETD($\beta$) found becomes smaller, and the learning also becomes slower. For LC-ETD1($\beta$) and LC-ETD2($\nu$) (Figures 4(d) and 4(e)), they learn faster with larger values of the decay parameter. On the other hand, LC-ETD3($\beta$) is not very sensitive to the value of $\beta$ (Figure 4(f)).

Figure 4(b) summarizes the above results. We can see that even in the high variance setting, LC-ETD instances are still better: They converge faster to the lowest error and are less sensitive to the decay parameter compared to ETD($\beta$).

## THE BIAS-VARIANCE TRADE-OFF

We now analyze the bias-variance trade-off that $\beta$ and $\nu$ control. We first analyze the bias and variance of the trace $F_t$ in Eq. (7) for different algorithms. Ideally, the expectation of $F_t$ given $S_t = s$ should be $\frac{\mathbb{P}_\pi(S_t=s)}{\mathbb{P}_\mu(S_t=s)}$, which converges to the density ratio in the limit and corrects the distribution of the update back to the on-policy distribution. Full-IS-TD achieves a zero bias but has the highest variance. LC-ETD instances have a relatively lower variance and a non-zero bias that will converge to zero asymptotically. On the other hand, Scaled ETD($\beta$) exhibits an even lower variance but a persistent bias. We conducted experiments on the Two-state task, sampling 100,000 trajectories of length 30 to estimate the bias and variance of $F_t$, as shown in Figure 5. Further details and discussions can be found in the appendix.

Figure 5 shows that increasing the decay parameter reduces the bias and increases the variance for all algorithms but with different speeds of change. LC-ETD1($\beta$) exhibits symmetric bias and variance curves, with the lowest variance and the highest bias at $\beta = 0$ (Off-policy TD), and the lowest bias and the highest variance at $\beta = 1$ (Full-IS-TD). Scaled ETD($\beta$) also connects Off-policy TD with Full-IS-TD, but it is not consistent and only becomes less biased as $\beta$ increases. It is also worth mentioning that its bias is persistent, while the bias of LC-ETD instances will fade away as more time steps are given. Additionally, LC-ETD2($\nu$) and LC-ETD3($\beta$) combined also form a polygonal line connecting Off-policy TD and Full-IS-TD. The bias and variance curves of the two algorithms combined form a similar shape to that of LC-ETD1($\beta$) but much wider. As a result, these two algorithms are less sensitive to the decay parameter but risk not achieving the best trade-off. Generally, LC-ETD2($\nu$) holds the best trade-off point in tasks with high variance.

We next look at how the updates at the two states are actually weighted in the experiment on the Two-state task. We calculate the ratio of $F_t$'s averages at the two states for every $1{,}000$ steps. Then we compute the absolute error of this ratio to the ratio of the density ratios at the two states as a measure of how effective $F_t$ is in reweighting the update. We refer to this error as the *ratio error* in the remaining text. We use the same data that generates Figure 2 and show the resulting ratio errors in logarithmic scale in Figure 6, which are averaged over $100$ runs. The shaded region near each curve represents the standard error, which is unnoticeable.

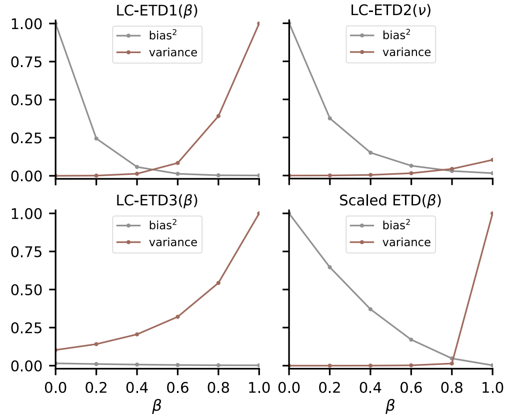


Figure 5: Bias-variance trade-off of different algorithms. The y-axis shows the normalized bias and variance of $F_t$.

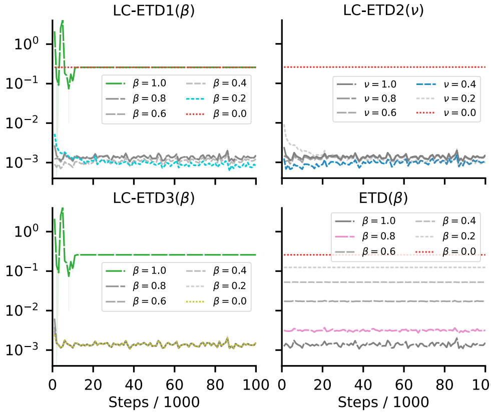


Figure 6: Ratio errors under different states in the experiment on the Two-state task. The y-axis shows the ratio error.

From Figure 6, we can see that the level of the ratio error has a positive correlation with $\overline{\text{RMSVE}}$ plotted in Figure 2. For LC-ETD instances with the decay parameter in the interval $[0.2, 0.8]$, their ratio errors are among the lowest. For Full-IS-TD (LC-ETD1($\beta$) and LC-ETD3($\beta$) with $\beta = 1$), its ratio error is very unstable at the beginning and then quickly remains at the same level as Off-policy TD (LC-ETD1($\beta$) with $\beta = 0$ and LC-ETD2($\nu$) with $\nu = 0$). This is because its $F_t$, the full IS-ratio product, diminishes to near zero after some steps, resulting in the ratio of $F_t$ remaining at 1 due to numerical issues. On the other hand, the ratio error of ETD($\beta$) becomes smaller and noisier as $\beta$ decreases. Noted that though the ratio error of ETD($\beta$) when $\beta = 1$ is also among the lowest, its performance is extremely unstable, as shown in Figure 2(c). This is because the magnitude of $F_t$ at both states is enormous despite the small ratio error.

In summary, the analysis illustrates how the decay parameters $\beta$ and $\nu$ affect the bias and variance of $F_t$, providing insights into the property of the corresponding algorithm.

## 5 CONCLUSIONS AND DISCUSSION

In this paper, we first introduced Average Emphatic TD (AETD($\lambda$)), a new consistent off-policy algorithm. To attain a smooth bias-variance trade-off, we unified AETD($\lambda$) with some existing algorithms (Precup, 2000; Precup et al., 2001; Sutton et al., 2016; Hallak et al., 2016). The resulting unified algorithm contains a new family of consistent algorithms, Loosely Consistent Emphatic TD (LC-ETD($\lambda$, $\beta$, $\nu$)), which has several desired theoretical and empirical properties. Firstly, different from ETD($\lambda$, $\beta$), LC-ETD($\lambda$, $\beta$, $\nu$) is guaranteed to be stable regardless of the values of its parameters. Secondly, while ETD($\lambda$, $\beta$) has a biased fixed point, LC-ETD($\lambda$, $\beta$, $\nu$) has the same fixed point as On-policy TD($\lambda$). Thirdly, the bias-variance trade-off that its parameters control makes LC-ETD($\lambda$, $\beta$, $\nu$) practical, providing an effective remedy to Full-IS-TD($\lambda$), the only consistent method previously. To our knowledge, LC-ETD($\lambda$, $\beta$, $\nu$) is the *first practical, consistent* algorithm for off-policy TD learning under general linear function approximation. By constraining LC-ETD($\lambda$, $\beta$, $\nu$)'s decay parameters, we obtained its three instances with the same number of parameters as ETD($\lambda$, $\beta$). Experiment results on a didactic example and a complex task with high variance showed a competitive performance of the instances, validating the effectiveness and practicality of LC-ETD($\lambda$, $\beta$, $\nu$).

Despite having the ability to control a smooth bias-variance trade-off, LC-ETD($\lambda$, $\beta$, $\nu$) still suffers from high variance issue to some degree (see the appendix). This issue is inherent to all importance-sampling-based methods including ETD($\lambda$, $\beta$) and Full-IS-TD($\lambda$). Potential treatments include periodically restarting or truncating the followon trace (Guan et al., 2022; Zhang and Whiteson, 2022) and learning an expected followon trace (Zhang et al., 2020c; Jiang et al., 2022). Investigating these and new approaches to further reduce the variance is one direction to be explored. Another unanswered question is the convergence of LC-ETD($\lambda$, $\beta$, $\nu$). Same as ETD($\lambda$) (Sutton et al., 2016), we have provided the stability guarantee of LC-ETD($\lambda$, $\beta$, $\nu$), which is an important necessary condition of its convergence. Similar to proving the convergence of ETD($\lambda$) (Yu, 2016), significant technical challenges may present in proving the convergence of LC-ETD($\lambda$, $\beta$, $\nu$). Thus, we leave it for future work.

### Acknowledgements

This research is supported by the Canada CIFAR AI Chairs program, the Reinforcement Learning and Artificial Intelligence (RLAI) laboratory, the Alberta Machine Intelligence Institute (Amii), and the Natural Sciences and Engineering Research Council (NSERC) of Canada. Jiamin He also gratefully acknowledges Richard S. Sutton and Huizhen Yu for valuable discussions and thanks the Digital Research Alliance of Canada for providing computational resources.

## References

Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A. C., and Bellemare, M. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29304–29320, 2021.

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

Chen, J. and Jiang, N. Offline reinforcement learning under value and density-ratio realizability: the power of gaps. In *Uncertainty in Artificial Intelligence*, pp. 378–388. PMLR, 2022.

Gelada, C. and Bellemare, M. G. Off-policy deep reinforcement learning by bootstrapping the covariate shift. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

Ghiassian, S. and Sutton, R. S. An empirical comparison of off-policy prediction learning algorithms in the four rooms environment. *arXiv preprint arXiv:2109.05110*, 2021.

Guan, Z., Xu, T., and Liang, Y. PER-ETD: A polynomially efficient emphatic temporal difference learning method. In *International Conference on Learning Representations*, 2022.

Hallak, A. and Mannor, S. Consistent on-line off-policy evaluation. In *International Conference on Machine Learning*, pp. 1372–1383. PMLR, 2017.

Hallak, A., Tamar, A., Munos, R., and Mannor, S. Generalized emphatic temporal difference learning: Bias-variance analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.

Huang, A., Chen, J., and Jiang, N. Reinforcement learning in low-rank mdps with density features. *arXiv preprint arXiv:2302.02252*, 2023.

Jiang, R., Zhang, S., Chelu, V., White, A., and van Hasselt, H. Learning expected emphatic traces for deep RL. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

Kolter, J. The fixed points of off-policy TD. In *Advances in Neural Information Processing Systems*, volume 24, 2011.

Lee, J., Jeon, W., Lee, B., Pineau, J., and Kim, K.-E. OptiDICE: Offline policy optimization via stationary distribution correction estimation. In *International Conference on Machine Learning*, pp. 6120–6130. PMLR, 2021.

Liu, Q., Li, L., Tang, Z., and Zhou, D. Breaking the curse of horizon: Infinite-horizon off-policy estimation. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Nachum, O., Chow, Y., Dai, B., and Li, L. DualDICE: Behavior-agnostic estimation of discounted stationary distribution corrections. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Precup, D. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, pp. 80, 2000.

Precup, D., Sutton, R. S., and Dasgupta, S. Off-policy temporal-difference learning with function approximation. In *International Conference on Machine Learning*, pp. 417–424, 2001.

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

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. MIT press, 2018.

Sutton, R. S., Precup, D., and Singh, S. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2): 181–211, 1999.

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

Sutton, R. S., Bowling, M. H., and Pilarski, P. M. The Alberta Plan for AI research. *arXiv preprint arXiv:2208.11173*, 2022.

Tsitsiklis, J. and Van Roy, B. Analysis of temporal-diffference learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 9, 1996.

Varga, R. *Matrix Iterative Analysis*. Springer Berlin Heidelberg, 1999.

White, A., Modayil, J., and Sutton, R. S. Scaling life-long off-policy learning. In *2012 IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, pp. 1–6. IEEE, 2012.

Yu, H. Weak convergence properties of constrained emphatic temporal-difference learning with constant and slowly diminishing stepsize. *Journal of Machine Learning Research*, 17(1):7745–7802, 2016.

Zhan, W., Huang, B., Huang, A., Jiang, N., and Lee, J. Offline reinforcement learning with realizability and single-policy concentrability. In *Conference on Learning Theory*, pp. 2730–2775. PMLR, 2022.

Zhang, R., Dai, B., Li, L., and Schuurmans, D. GenDICE: Generalized offline estimation of stationary values. In *International Conference on Learning Representations*, 2020a.

Zhang, S. and Whiteson, S. Truncated emphatic temporal difference methods for prediction and control. *Journal of Machine Learning Research*, 23(153):1–59, 2022.

Zhang, S., Liu, B., and Whiteson, S. GradientDICE: Rethinking generalized offline estimation of stationary values. In *International Conference on Machine Learning*, pp. 11194–11203. PMLR, 2020b.

Zhang, S., Liu, B., Yao, H., and Whiteson, S. Provably convergent two-timescale off-policy actor-critic with function approximation. In *International Conference on Machine Learning*, pp. 11204–11213. PMLR, 2020c.