Learning Expected Emphatic Traces for Deep RL

Ray Jiang[1], Shangtong Zhang[2], Veronica Chelu[3], Adam White[1, 4], Hado van Hasselt[1]
[1] DeepMind, London, UK
[2] University of Oxford, Oxford, UK
[3] McGill University, Montreal, QC, Canada
[4] Amii, Department of Computing, Science, University of Alberta
rayjiang@deepmind.com, shangtong.zhang@cs.ox.ac.uk, veronica.chelu@mail.mcgill.ca, hado@deepmind.com, adamwhite@deepmind.com

Abstract

Off-policy sampling and experience replay are key for improving sample efficiency and scaling model-free temporal difference learning methods. When combined with function approximation, such as neural networks, this combination is known as the deadly triad and is potentially unstable. Recently, it has been shown that stability and good performance at scale can be achieved by combining emphatic weightings and multi-step updates. This approach, however, is generally limited to sampling complete trajectories in order, to compute the required emphatic weighting. In this paper we investigate how to combine emphatic weightings with non-sequential, off-line data sampled from a replay buffer. We develop a multi-step emphatic weighting that can be combined with replay, and a time-reversed n-step TD learning algorithm to learn the required emphatic weighting. We show that these state weightings reduce variance compared with prior approaches, while providing convergence guarantees. We tested the approach at scale on Atari 2600 video games, and observed that the new X-ETD(n) agent improved over baseline agents, highlighting both the scalability and broad applicability of our approach.

Many deep reinforcement learning systems are not sample efficient. A simple and effective way to improve sample efficiency is to make better use of prior experience via replay (Lin 1992; Mnih et al. 2015; Schaul et al. 2016; Hessel et al. 2018). Previous work demonstrated, somewhat surprisingly, that increasing the amount of replay in a model-free learning system can surpass the sample efficiency and final performance of model-based agents which utilize significantly more computation (van Hasselt, Hessel, and Aslanides 2019).

While improving on sample efficiency, using experience replay also introduces more potential for instability. Most approaches update from mini-batches of previous experience corresponding to older policies, and is therefore off-policy (Mnih et al. 2015; Hessel et al. 2018). Unfortunately combining bootstrapping via temporal-difference updates, function approximation and off-policy learning—known as the *deadly triad* (Sutton and Barto 2018)—can destabilize learning resulting in "soft divergence", slower learning, and reduced sample efficiency even if the parameters do not fully diverge (van Hasselt et al. 2018). Additionally, learning methods based on off-policy *importance sampling* (IS) corrections can result in high variance and poor performance during learning. This can be improved in practice by bootstrapping more, for instance by cleverly clipping the IS ratios as in the V-trace algorithm (Espeholt et al. 2018) or ABTD (Mahmood, Yu, and Sutton 2017), though bootstrapping too much can exacerbate issues related to the deadly triad.

In order to prevent divergence, we can try to correct the mismatch between the state distribution in the replay buffer and the current policy. The emphatic TD(λ) or ETD(λ) algorithm (Sutton, Mahmood, and White 2016) reweights the TD(λ) updates with an "emphatic" state weighting based on a "followon" trace that, intuitively, keeps track of how important each state is in the learning process. For instance, states that are heavily used to update other state values, via bootstrapping, will receive more emphasis, which ensures their values are sufficiently accurate even if they are updated infrequently. This prevents divergent learning dynamics.

ETD(λ) uses eligibility traces and has not yet been combined with neural network function approximation or replay. Fortunately, the idea of emphatic weighting is not restricted to trace-based ("backward-view") algorithms and can be extended to other settings. For instance, n*-step Emphatic TD* (NETD) (Jiang et al. 2021) is a recent algorithm that combines emphatic weighting with n-step forward-view updates as well as V-trace learning targets. For consistency with the canonical name TD(n), for n-step TD learning, we call this algorithm ETD(n) in this paper. This was shown to outperform V-trace at scale in Atari and diagnostic MDP experiments (Jiang et al. 2021).

The emphatic weightings used in ETD(n) are sequentially accumulated over time, in the form of trajectory-dependent traces, and can thus only be computed from online sequential trajectories, or full episodes of offline trajectories. In this paper we investigate how to combine emphatic weightings with non-sequential, off-line data sampled from a replay buffer. The idea is to estimate *expected* emphatic weightings as a function of state (Zhang et al. 2020; van Hasselt et al. 2020), allowing us to appropriately weight the learning updates even if the inputs are sampled out of order. This reduces well-known variance issues with emphatic weightings (Ghiassian et al. 2018; Imani, Graves, and White 2018; Zhang et al. 2020). We show in Sec. that well-estimated emphatic weights reduce the potentially high variance of ETD(n) and achieve convergence with an upper bound on the bias from the ground-truth value function.

Our contributions include 1) an off-policy time-reversed TD learning algorithm to learn the expected n-step emphatic trace using non-sequential data; 2) a discussion of potential stabilization techniques; 3) an analysis theoretical properties of variance, stability and convergence for the resulting algorithm X-ETD(n); 4) an investigation of practical benefits of the approach when used at scale: we observed that X-ETD(n) outperformed the baseline on Atari when using replay.

Background

We denote random variables with uppercase (e.g., S) and the obtained values with lowercase letters (e.g., $S = s$). Multi-dimensional functions or vectors are bolded (e.g., $\mathbf{b}$), as are matrices (e.g. $\mathbf{A}$). For all state-dependent functions, we also allow time-dependent shorthands (e.g., $\gamma_t = \gamma(S_t)$).

Reinforcement Learning Problem Setup

We consider the usual RL setting in which an agent interacts with an environment, modelled as an infinite horizon *Markov Decision Process* (MDP) $(\mathcal{S}, \mathcal{A}, P, r)$, with a finite state space $\mathcal{S}$, a finite action space $\mathcal{A}$, a state-transition distribution $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ (with $\mathcal{P}(\mathcal{S})$ the set of probability distributions on $\mathcal{S}$ and $P(s'|s,a)$ the probability of transitioning to state s' from s by choosing action a), and a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. A policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ maps states to distributions over actions; $\pi(a|s)$ denotes the probability of choosing action a in state s and $\pi(s)$ denotes the probability distribution of actions in state s. Let S_t, A_t, R_t denote the random variables of state, action and reward at time t, respectively.

The goal of *policy evaluation* is to estimate the *value function* v_π, defined as the expectation of the discounted return under policy π:

$$G_t \doteq R_{t+1} + \Sigma_{i=t+1}^{\infty} \gamma_i R_{i+1} = R_{t+1} + \gamma_{t+1} G_{t+1} \,, \tag{1}$$

$$v_\pi(s) \doteq \mathbb{E}_{A_k \sim \pi(S_k), S_{k+1} \sim P(S_k, A_k)\ \forall k \geq t}[G_t \mid S_t = s] \,, \tag{2}$$

where $\gamma : \mathcal{S} \to [0, 1]$ is a discount factor. We consider function approximation and use $v_\mathbf{w}$ as our estimate of v_π, where $\mathbf{w}$ are parameters of $v_\mathbf{w}$ to be updated.

In the case of *off-policy* policy evaluation, though our goal is to estimate v_π, the actions for interacting with the MDP are sampled according to a different policy μ. We refer to π and μ as target and behavior policies respectively and make the following assumption for the behavior policy μ:

Assumption 1. *(Ergodicity) The Markov chain induced by μ is ergodic.*

Assumption 2. *(Coverage) $\pi(a|s) > 0 \implies \mu(a|s) > 0$ holds for any (s, a).*

Under Assumption 1, we use d_μ to denote the ergodic distribution of the chain induced by μ. In this paper, we consider two off-policy learning settings: the *sequential setting* and the *i.i.d. setting*. In the sequential setting, the algorithm is presented with an infinite sequence as induced by the interaction

$$(S_0, A_0, R_1, S_1, A_1, R_2, \dots),$$

where $A_t \sim \mu(S_t), R_{t+1} \doteq r(S_t, A_t), S_{t+1} \sim P(S_t, A_t)$. The idea is then that we update the value and/or policy at each of these states S_t, using data following the state (e.g., the sampled return). Updates at state S_t always happen before updates at states S_{t+k}, for $k > 0$.

In the i.i.d. setting, the algorithm is presented with an infinite number of *finite* sequences of length n

$$\{(S_0^k, A_0^k, R_1^k, S_1^k, A_1^k, R_2^k, \dots, S_n^k)\}_{k=1,2,\dots} \,,$$

where the starting state of a sequence is sampled i.i.d., such that $S_0^k \sim d_\mu$, and then the generating process for the subsequent steps is the same as before: $A_t^k \sim \mu(S_t^k), R_{t+1}^k \doteq r(S_t^k, A_t^k), S_{t+1}^k \sim P(S_t^k, A_t^k)$. The idea is then that we update the value and/or policy of the first state in each sequence, S_0^k, using the rest of that sequence, e.g., by constructing a bootstrapped n-step return.

The sequential setting corresponds to the canonical agent-environment interaction (Sutton and Barto 2018). Sequential algorithms are often data inefficient, since typically each state S_t is updated only once and then discarded (e.g., Watkins and Dayan 2004). One way to improve data efficiency is to store transitions in a replay buffer (Lin 1992) and reuse these for further updates. If μ is stationary and these tuples are uniformly sampled from a large-enough buffer, their distribution is similar to d_μ. Hence uniform replay is akin to the i.i.d. setting. If we sample from the replay buffer with different priorities, e.g., S_0^k is sampled from some other distribution d_p, the updates to S_0^k can be reweighted with importance-sampling ratios $d_\mu(S_0^k)/d_p(S_0^k)$ to retain the similarity to the i.i.d. setting. Therefore, for simplicity and clarity, we present our theoretical results in the i.i.d. setting.[1]

Policy Evaluation

We use the sequential setting and linear function approximation to demonstrate three algorithms for off-policy evaluation. We denote the features of state S_t by $\boldsymbol{\phi}(S_t)$ or $\boldsymbol{\phi}_t$.

Off-policy TD(n) Off-policy TD(n) updates $\mathbf{w}$ as

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha \textstyle\sum_{k=t}^{t+n-1} \left(\prod_{i=t}^{k-1} \rho_i \gamma_{i+1} \right) \rho_k \delta_k(\mathbf{w}_t) \boldsymbol{\phi}_t, \tag{3}$$

where $\rho_t \doteq \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)}$ is an importance sampling (IS) ratio and $\delta_k(\mathbf{w}_t)$ is the TD error:

$$\delta_k(\mathbf{w}_t) = R_{k+1} + \gamma_{k+1} v_{\mathbf{w}_t}(S_{k+1}) - v_{\mathbf{w}_t}(S_k) \,. \tag{4}$$

ETD(n) Off-policy TD(n) can possibly diverge with function approximation. Emphatic weightings are an approach to address this issue (Sutton, Mahmood, and White 2016). In

[1] In practice, computing $d_\mu(S_0^k)/d_p(S_0^k)$ exactly is usually impossible. One can, however, approximate it with $1/(d_p(S_0^k)N)$ with N being the size of the replay buffer. We refer the reader to (Schaul et al. 2016) for more details about this approximation.

particular, ETD(n) considers the following "followon trace" to stabilize the off-policy TD(n) updates (Jiang et al. 2021):

$$F_t = \left(\prod_{i=t-n}^{t-1} \rho_i \gamma_{i+1}\right) F_{t-n} + 1, \tag{5}$$

with $F_0 = F_1 = \cdots = F_{n-1} = 1$, thus updating $\mathbf{w}_t$ iteratively as

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha F_t \sum_{k=t}^{t+n-1} \left(\prod_{i=t}^{k-1} \rho_i \gamma_{i+1}\right) \rho_k \delta_k(\mathbf{w}_t) \boldsymbol{\phi}_t. \tag{6}$$

Jiang et al. (2021) proved that this ETD(n) update is stable. In this paper, we consider *stability* in the sense of Sutton, Mahmood, and White (2016): a stochastic algorithm computing $\{\mathbf{w}_t\}$ according to $\mathbf{w}_{t+1} \doteq \mathbf{w}_t + \alpha_t(\mathbf{b}_t - \mathbf{A}_t\mathbf{w}_t)$ is *stable* if $\mathbf{A} \doteq \lim_{t\to\infty} \mathbb{E}[\mathbf{A}_t]$ is positive definite (p.d.)[2].

Notice F_t in (5) is a trace, defined on a sequence of transitions ranging, via recursion on previous values F_{t-n}, into the indefinite past. When we do not have access to this sequence to compute the right weighting for a given state, for instance because we sampled this state uniformly from a replay buffer, we need to consider an alternative way to correctly weight the update. **This incompatibility between the i.i.d. setting and ETD(n) is the main problem we address in the paper**. To differentiate from our proposed emphatic weighting, from here on we refer to the ETD(n) trace as the *Monte Carlo* trace since it is a Monte Carlo return in reversed time with "reward" signal of 1 every n steps. In addition, we use the term *emphasis* as a shorthand for "emphatic trace".

Proposal: Learn the Expected Emphasis

In order to apply emphatic traces to non-sequential i.i.d. data, we propose, akin to Zhang et al. (2020), to directly learn a prediction model that estimates the limiting expected emphatic trace, i.e., we train a function f_θ parameterized by θ such that $f_\theta(s)$ approximates $\lim_{k\to\infty} \mathbb{E}_\mu[F_k \mid S_k = s]$.[3] We use the learned emphasis f_{θ_k} in place of the Monte Carlo ETD(n) trace F_k, to re-weight the n-step TD update in (6). We refer to the resulting value learning algorithm *eXpected Emphatic TD(n), X-ETD(n)*. Thanks to the trace prediction model f_θ, a sequential trajectory is no longer necessary for computing the emphatic weighting X-ETD(n). We know that using a learned expected emphasis can introduce approximation errors. Thus Section contains a theoretical analysis, which shows that as long as the function approximation error is not too large, stability, convergence, and a reduction in variance, are all guaranteed. We dedicate this section to the describing how to learn f_θ.

The n-step emphatic traces in (5) are designed to emphasize n-step TD updates. Consequentially, the trace recursion in (5) follows the same blueprint as TD(n), but in the reverse direction of time. Hence a natural choice of learning algorithm for the expected emphasis is *time-reversed TD learning* that learns the "reward" 1 every n^{th} step using off-policy TD(n) with the time index reversed. Considering the i.i.d. setting, we update θ_k using a "semi-gradient" (Sutton and Barto 2018) TD update:

$$\theta_{k+1} = \theta_k + \tag{7}$$
$$\alpha_k^\theta \left[\left(\prod_{t=1}^{n} \gamma_t^k \rho_{t-1}^k\right) f_{\theta_k}(S_0^k) + 1 - f_{\theta_k}(S_n^k)\right] \nabla_{\theta_k} f_{\theta_k}(S_n^k),$$

where α_k^θ is a possibly time-dependent step size, $\rho_t^k \doteq \frac{\pi(A_t^k|S_t^k)}{\mu(A_i^k|S_t^k)}$, and $\gamma_t^k \doteq \gamma(S_t^k)$. This update corresponds to the semi-gradient of the following loss,

$$\mathcal{L}_k^F \doteq \left[\left(\prod_{t=1}^{n} \gamma_t^k \rho_{t-1}^k\right) f_{\theta_k}(S_0^k) + 1 - f_{\theta_k}(S_n^k)\right]^2. \tag{8}$$

For the case of linear value function approximation, the update in (7) may not be stable because its update matrix, $\mathbf{\Phi}^T(\mathbf{I} - (\mathbf{\Gamma}\mathbf{P}_\pi^T)^n)\mathbf{D}_\mu\mathbf{\Phi}$, is not guaranteed to be positive definite (details in the appendix). Here $\mathbf{P}_\pi$ is the transition matrix such that $\mathbf{P}_\pi(s,s') \doteq \sum_a \pi(a|s)P(s'|s,a)$, $\mathbf{\Gamma}$ is a diagonal matrix such that $\mathbf{\Gamma}(s,s) \doteq \gamma(s)$, $\mathbf{D}_\mu$ is a diagonal matrix whose diagonal entry is d_μ, and $\mathbf{\Phi}$ is the feature matrix whose s-th row is $\boldsymbol{\phi}(s)^\top$. Thus we propose two stabilization techniques for the time-reversed TD learning updates.

IS Clipping One straightforward way is to clip the IS ratios in (7) just like in V-trace (Espeholt et al. 2018), i.e., we update θ iteratively as

$$\theta_{k+1} = \theta_k + \nabla_{\theta_k} f_{\theta_k}(S_n^k) \times \tag{9}$$
$$\alpha_k^\theta \left[\left(\prod_{t=1}^{n} \gamma_t^k \min(\rho_{t-1}^k, \bar\rho)\right) f_{\theta_k}(S_0^k) + 1 - f_{\theta_k}(S_n^k)\right],$$

for some $\bar\rho > 0$; typically $\bar\rho = 1$.

Define the substochastic matrix $\mathbf{P}_{\bar\rho}$ such that for any state s, s',

$$\mathbf{P}_{\bar\rho}(s,s') \doteq \sum_a \mu(a|s) \min(\rho(a|s), \bar\rho) p(s'|s,a)\gamma(s'). \tag{10}$$

Then the update matrix of (9) is $\mathbf{\Phi}^T(\mathbf{I} - (\mathbf{P}_{\bar\rho}^T)^n)\mathbf{D}_\mu\mathbf{\Phi}$ (see details in the appendix).

We prove that when estimating the expected emphasis using linear function approximation, there exist conditions under which we can guarantee stability at the cost of incurring additional bias.

Proposition 1. *There exists a constant $\tau > 0$ such that the update in* (9) *is stable whenever $\bar\rho < \tau$.*

See its proof in the appendix. One such constant is $\tau = \max_s 1/\gamma(s)$ where the maximum of discounts $\gamma(s)$ is over states. Notice that while achieving stability, clipping at $1/\gamma$ also restricts variance of learning to a finite amount since the Monte Carlo ETD(n) trace is bounded. In practice, we tune $\bar\rho$ to optimize a bias-stability trade-off.

Auxiliary Monte-Carlo loss In most learning settings (e.g., (Mnih et al. 2015)), both sequential samples and i.i.d. samples are available. To take advantage of this fact, we can stabilize the emphasis learning by partially regressing on the

[2] See (13) for specific forms of vector b_t, matrix A_t in our case.

[3] For the ease of presentation, we assume the existence of the limit, following Sutton, Mahmood, and White (2016); Jiang et al. (2021). The existence can be proven similar to Lemma 1 of Zhang, Boehmer, and Whiteson (2019), which we leave for future work.

Monte Carlo emphatic trace. We can thus learn the parameters θ by TD-learning using samples from the replay buffer and by Monte Carlo learning using online experience:

$$\theta_{k+1} = \theta_k + \alpha_k^\theta \beta \left(F_k - f_{\theta_k}(S_k)\right) \nabla_{\theta_k} f_{\theta_k}(S_k) + \tag{11}$$
$$\alpha_k^\theta \left[\left(\prod_{t=1}^{n} \gamma_t^k \rho_{t-1}^k \right) f_{\theta_k}(S_0^k) + 1 - f_{\theta_k}(S_n^k) \right] \nabla_{\theta_k} f_{\theta_k}(S_n^k).$$

This update corresponds to the joint loss function:

$$\mathcal{L}_k^{F,\mathrm{MC}} \doteq \left[\left(\prod_{t=1}^{n} \gamma_t^k \rho_{t-1}^k \right) f_{\theta_k}(S_0^k) + 1 - f_{\theta_k}(S_n^k) \right]^2 + \beta (f_{\theta_k}(S_k) - F_k)^2, \tag{12}$$

where β is a hyper-parameter for balancing the two losses. When f_θ uses linear function approximation, we prove the following guarantee on its stability (proof in the appendix).

Proposition 2. *There exists a constant ξ such that the update in* (11) *is stable whenever $\beta > \xi$.*

The time-reversed TD update can be unstable, whereas the Monte Carlo update target F_k can have large variance (Sutton, Mahmood, and White 2016; Jiang et al. 2021). By choosing β, we optimize a variance-stability trade off.

Expected Emphatic TD Learning

To prevent deadly triads, we use the learned expected emphasis f_θ to re-weight the learning updates of TD(n). In this section, we analyze the resulting algorithm, X-ETD(n). For simplicity, let the trace model f_θ be parameterized by a fixed parameter θ.[4] In this section, we analyze X-ETD(n) in the sequential setting for the ease of presentation. A similar analysis would apply to X-ETD(n) in the i.i.d. setting. Then X-ETD(n) updates $\mathbf{w}$ iteratively as

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha_t^{\mathbf{w}} f_\theta(S_t) \Delta_t^{\mathbf{w}}, \tag{13}$$

where

$$\Delta_t^{\mathbf{w}} \doteq \textstyle\sum_{k=t}^{t+n-1} (\prod_{i=t}^{k-1} \gamma_{i+1}\rho_i) \rho_k (R_{k+1} + \gamma_{k+1} \mathbf{w}_t^\top \boldsymbol{\phi}(S_{k+1}) - \mathbf{w}_t^\top \boldsymbol{\phi}(S_t)) \boldsymbol{\phi}(S_t).$$

Equivalently, we can write (13) as

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \alpha_t^{\mathbf{w}} (\mathbf{b}_t - \mathbf{A}_t \mathbf{w}_t), \quad \text{where} \tag{14}$$
$$\mathbf{A}_t \doteq f_\theta(S_t) \boldsymbol{\phi}(S_t) \textstyle\sum_{k=t}^{t+n-1} \left(\prod_{i=t}^{k-1} \gamma_{i+1}\rho_i \right) \rho_k [\boldsymbol{\phi}(S_k) - \gamma_{k+1} \boldsymbol{\phi}(S_{k+1})]^\top \tag{15}$$
$$\mathbf{b}_t \doteq f_\theta(S_t) \textstyle\sum_{k=t}^{t+n-1} \left(\prod_{i=t}^{k-1} \gamma_{i+1}\rho_i \right) \rho_k R_{k+1} \boldsymbol{\phi}(S_t) \tag{16}$$

As we use $f_\theta(S_t)$ to reweight the update, it is convenient to define a diagonal matrix $\mathbf{D}_\mu^\theta$ with diagonal entries $[\mathbf{D}_\mu^\theta]_{ss} \doteq d_\mu(s) f_\theta(s)$ for any state s. In X-ETD(n), we approximate $\lim_{t\to\infty} \mathbb{E}_\mu[F_t | S_t = s]$ with $f_\theta(s)$. In this, we also define a ground-truth diagonal matrix $\mathbf{D}_\mu^f$ such that $[\mathbf{D}_\mu^f]_{ss} \doteq d_\mu(s) \lim_{t\to\infty} \mathbb{E}_\mu[F_t | S_t = s]$, and their difference, $\mathbf{D}_\mu^\epsilon \doteq \mathbf{D}_\mu^\theta - \mathbf{D}_\mu^f$, is the ($d_\mu$-weighted) function approximation error matrix of the emphasis approximation. It can be computed that

$$\mathbf{A} \doteq \lim_{t\to\infty} \mathbb{E}[\mathbf{A}_t] = \boldsymbol{\Phi}^\top \mathbf{D}_\mu^\theta (\mathbf{I} - (\mathbf{P}_\pi \boldsymbol{\Gamma})^n) \boldsymbol{\Phi}, \tag{17}$$
$$\mathbf{b} \doteq \lim_{t\to\infty} \mathbb{E}[\mathbf{b}_t] = \boldsymbol{\Phi}^\top \mathbf{D}_\mu^\theta \mathbf{r}_\pi^n, \tag{18}$$

where $\mathbf{r}_\pi^n \doteq \sum_{i=0}^{n-1} (\mathbf{P}_\pi \boldsymbol{\Gamma})^i \mathbf{r}_\pi$ is the n-step reward vector with $\mathbf{r}_\pi(s) \doteq \sum_a \pi(a|s) r(s,a)$.

Variance

Learning to estimate the emphatic trace not only makes value-learning compatible with offline learning methods that make use of replay buffers, but is also instrumental in reducing the variance of the value-learning updates. The incremental update of ETD(n) in (6) can be rewritten as $F_t \Delta_t^{\mathbf{w}}$. The following proposition shows that when the trace approximation error is small enough, variance in learning can indeed be reduced by replacing the Monte Carlo trace F_t with the learned trace $f_\theta(S_t)$.

Proposition 3. *(Reduced variance) Let $\epsilon_s \doteq |f_\theta(s) - f(s)|$ be the trace approximation error at a state s. For any s, there exists a time $\bar{t} > 0$, such that for all $t > \bar{t}$,*

$$\epsilon_s(\epsilon_s + 2f(s)) < \mathbb{V}(F_t | S_t = s) \implies \mathbb{V}(f_\theta(S_t)\Delta_t^{\mathbf{w}} | S_t = s) \le \mathbb{V}(F_t \Delta_t^{\mathbf{w}} | S_t = s). \tag{19}$$

The inequality is strict if $\mathbb{V}(\Delta_t^{\mathbf{w}} | S_t = s) > 0$.

(Proof in the appendix.) In some cases $\mathbb{V}(F_t | S_t = s)$ can be infinite (Sutton, Mahmood, and White 2016); then the condition in Proposition 3 holds trivially. This also underpins the importance of variance reduction.

Convergence

Next, under the following assumption about the learning rate, we show the convergence of (13).

Assumption 3. *(Learning rates) The learning rates $\{\alpha_t^{\mathbf{w}}\}_{t=0}^\infty$ are nonnegative, deterministic, and satisfy $\sum_t \alpha_t^{\mathbf{w}} = \infty, \sum_t (\alpha_t^{\mathbf{w}})^2 < \infty$.*

Theorem 1. *(Convergence of X-ETD(n)) Under Assumptions 1-3, for the iterates $\{\mathbf{w}_t\}$ generated by* (13)*, there exists a constant $\eta > 0$ such that*

$$\left\| \mathbf{D}_\mu^\epsilon \right\| < \eta \implies \lim_{t\to\infty} \mathbf{w}_t = \mathbf{A}^{-1}\mathbf{b} \quad a.s.. \tag{20}$$

The proof of this theorem is in the appendix, along with a stability guarantee for the X-ETD(n) updates. Theorem 1 shows that under some mild conditions, assuming the function approximation error is not too large, X-ETD(n) converges to $\mathbf{w}_\infty \doteq \mathbf{A}^{-1}\mathbf{b}$.. We now study the performance of $\mathbf{w}_\infty$, i.e., the distance between the value prediction by $\mathbf{w}_\infty$ and the true value function $\mathbf{v}_\pi$.

Proposition 4. *(Suboptimality of the fixed point) Under Assumptions 1 & **??**, there exists positive constants c_1, c_2, and c_3 such that*

$$\left\| \mathbf{D}_\mu^\epsilon \right\| \le c_1 \implies \left\| \boldsymbol{\Phi}\mathbf{w}_\infty - \mathbf{v}_\pi \right\| \le c_2 \left\| \mathbf{D}_\mu^\epsilon \right\| + c_3 \left\| \Pi_{\mathbf{D}_\mu^f} \mathbf{v}_\pi - \mathbf{v}_\pi \right\|_{\mathbf{D}_\mu^f}, \tag{21}$$

[4] This is not a special setting since the expected emphasis learning process is independent from learning parameters $\mathbf{w}$.

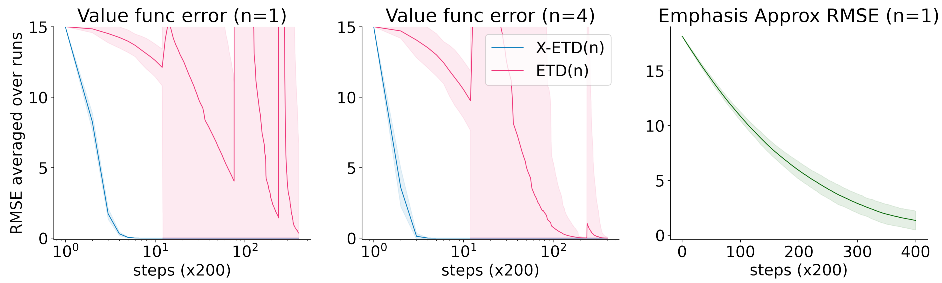


Figure 1: RMSE in the value estimates and RMSE in expected trace approximation over time in a modified version of Baird's counterexample (Baird 1995). We report the performance of each algorithm using the best performing hyper-parameters (according to RMSE of the value function) from an extensive sweep (described in text). Shaded regions indicate two standard deviations of the mean performance computed from 100 independent runs.

where $\left\|\Pi_{\mathbf{D}_\mu^f}\mathbf{v}_\pi - \mathbf{v}_\pi\right\|_{\mathbf{D}_\mu^f}$ is the value estimation error of the unbiased fixed point using the Monte Carlo emphasis. We prove this proposition in the appendix.

Illustration on Baird's Counterexample

We illustrate the theoretical results in this section on a small MDP based on Baird's counterexample (Baird 1995). The MDP has seven states with linear features. The over-parametrized features are designed to cause instability even though the true values can be represented. See Sutton and Barto (2018) for an extensive discussion and analysis of Baird's counterexample. As in Zhang et al. (2020) we modify the MDP (see Fig. **??** in the appendix), using a discount of $\gamma = 0.95$ and a target policy $\pi(\text{solid}|\cdot) = 0.3$. We tested all combinations of $\alpha^{\mathbf{w}} \in \{2^i : i = -6, \ldots, -14\}$ and $\alpha^\theta = \alpha^{\mathbf{w}}\beta$, with $\beta \in \{0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0, 2.0, 5.0\}$. Fig. 1 contains a summary; additional results in the appendix.

X-ETD(n) was more stable in this small but challenging MDP. ETD(n) exhibited high variance and instability. X-ETD(n) had very low variance, echoing the conclusion of Prop. 3 that X-ETD(n) has lower variance when its emphasis errors are small. X-ETD(n) also converged faster to the true fixed point, illustrating Theorem 1 and, moreover, achieving the optimal fixed point, far better than the worst case upper bound of Prop. 4. Note, even when choosing $\alpha^{\mathbf{w}}$ and β of X-ETD(n) to minimize the RMSE in the value function, the emphasis approximation error exhibits steady improvement.

Experiments

Back to the problem that motivated this paper, our goal is to stabilize learning at scale when using experience replay. Inspired by the performance achieved by Surreal (Jiang et al. 2021) and StacX (Zahavy et al. 2020), both extensions of IMPALA (Espeholt et al. 2018), we adopt the same off-policy setting of learning auxiliary tasks to test X-ETD(n), with the additional use of experience replay. It has become conventional to clip IS ratios to reduce variance and improve learning results (Espeholt et al. 2018; Zahavy et al. 2020; Hessel et al. 2021a). We similarly adapt X-ETD(n) to the control setting by clipping IS ratios at 1 in both policy evaluation, as described in Sec. , and applying the learned emphatic weighting to the corresponding policy gradients. Further details are in the appendix.

Data We evaluate X-ETD(n) on a widely used deep RL benchmark, Atari games from the Arcade Learning Environment (Bellemare et al. 2013)[5]. The input observations are in RGB format without downsampling or gray scaling. We use an action repeat of 4, with max pooling over the last two frames and the life termination signal. This is the same data format as that used in Surreal (Jiang et al. 2021) and StacX (Zahavy et al. 2020). In addition, we randomly sample half of the training data from an experience replay buffer which contains the most recent 10,000 sequences of length 20. In order to compare with previous works, we use the conventional 200M online frames training scheme, with an

[5]Licensed under GNU General Public License v2.0.

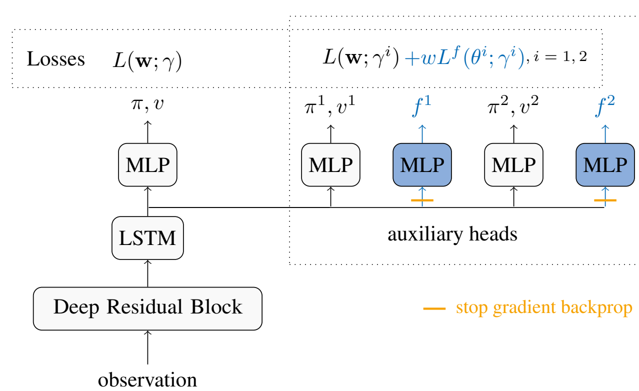


Figure 2: Block diagram of X-ETD(n). Agent has one main head, two auxiliary heads and two trace heads with stop gradients. The subset of architecture in gray denotes the baseline agent Surreal, and those highlighted in blue are used in trace learning. We use the IMPALA loss on each head with different discounts $\gamma, \gamma^1, \gamma^2$. The behavior policy is fixed to be π. The predicted traces f^1, f^2 are learned with time-reversed TD losses $L^f(\theta^i; \gamma^i)$ weighted by w for the auxiliary heads $i = 1, 2$.

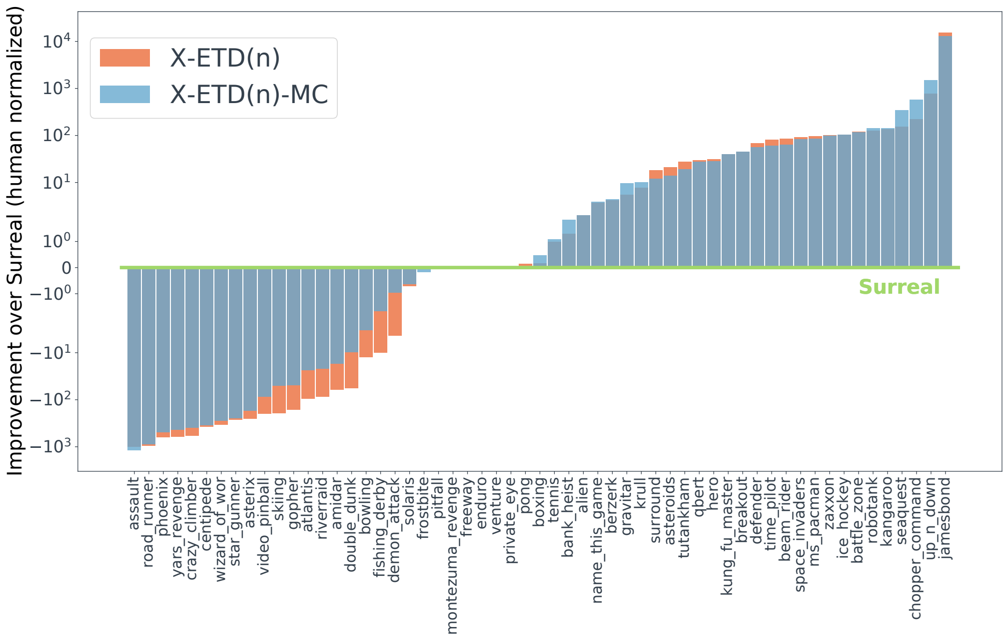


Figure 3: Improvement in individual human normalized game scores compared to Surreal: X-ETD(n) in orange and X-ETD(n)-MC in blue. Both improve over baseline Surreal in median human normalized scores on 57 Atari games, and scores improved in more games than in which they deteriorated. Results averaged between 200M - 250M frames (evaluation phase) across 3 seeds.

evaluation phase at 200M-250M learning frames.

Baseline Agent Surreal is an IMPALA-based agent that learns two auxiliary tasks with different discounts γ^1, γ^2 simultaneously while learning the main task (Fig. 2, in gray). The auxiliary tasks are learned off-policy since the agent generates behaviours only from the main policy output. Prior to applying X-ETD(n), we swept extensively on its hyperparameters to produce the best baseline we could find with 50% replay data (see the appendix for further details).

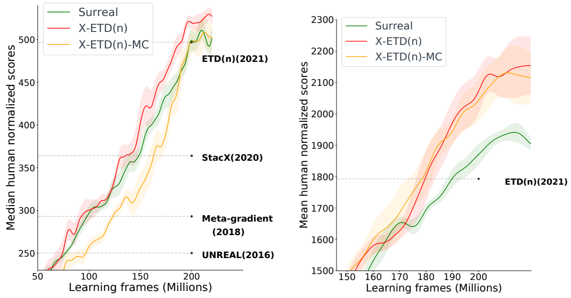
X-ETD(n) Agents We investigate whether X-ETD(n) updates can improve off-policy learning of the auxiliary tasks. For each of the two auxiliary tasks, we implement an additional Multilayer Perceptron (MLP) that predicts the expected emphatic trace using the time-reversed TD learning loss L^f in (8) (see Fig. 2, in blue). The prediction outputs f^i are then used to re-weight both the V-trace value and policy updates for the auxiliary task $i = 1, 2$, similar Jiang et al. (2021). In order to isolate the effect of using X-ETD(n) from any changes to internal representations as a result of the additional trace learning losses, we prevent the gradients from back-propagating to the core of the agent. We denote learned emphasis with auxiliary Monte Carlo loss as X-ETD(n)-MC, described in Sec. . We implement all agents in a distributed system based on JAX libraries (Hennigan et al. 2020; Budden et al. 2020; Hessel et al. 2020)[6] using a TPU Pod infrastructure called Sebulba (Hessel et al. 2021b).

Evaluation Running many seeds on all 57 Atari games is expensive. However a single metric with few seeds can be noisy, or hard to properly interpret. Hence we adopt a four-faceted evaluation strategy. We report mean and median human normalized training curves with standard deviations across 3 seeds (Fig. 4), accompanied by a bar plot of per-game improvements in normalized scores averaged across 3 seeds and the evaluation window (Fig. 3). In addition, to test rigorously whether X-ETD(n) improved performance, we apply a one-sided *Sign Test* (Arbuthnot 1712) on independent pairs of agent scores on 57 (games) x 3 (seeds) to compute its p-value, where scores are averaged across the evaluation window and the baseline and test agent seeds are paired randomly. To guarantee random pairing, we uniformly sample and pair the seeds 10,000 times and take the average number of games on which the test agent is better. The p-value is the probability of observing the stated results under the null hypothesis that the algorithm performs equally. Results might be thought of as statistically significant when $p < 0.05$.

Statistics	X-ETD(n)	X-ETD(n)-MC	Surreal
Median	503%	**537%**	525%
Mean	**2122%**	2090%	1879%
games > baseline	**97** (of 171)	**97** (of 171)	N/A
p-value	**0.046**	**0.046**	N/A

Table 1: Performance statistics for Surreal and learned emphases applied to Surreal on 57 Atari games. Scores are human normalized, averaged across 3 seeds and across the evaluation phase (200M-250M). Mean and median refer to human-normalized scores.

[6]Licensed under Apache License 2.0.



(a) Median scores (b) Mean scores

Figure 4: Training curves of (a) median and (b) mean human normalized scores on 57 Atari games, with standard deviations (shaded area) across 3 seeds.

Results The mean and median scores in Table 1 are human-normalized and averaged across 3 seeds, then averaged over 200-250M frames evaluation window. On a per-game level, both X-ETD(n) and X-ETD(n)-MC outperformed Surreal on 97 out of 57x3=171 games with a p-value of 0.046, as well as reaching higher mean scores. Fig. 3 shows per-game improvements over Surreal. The improvements are stable across two emphatic variants on the majority of games. In the games where learned emphasis hurt performance, we observed two scenarios: 1) the predicted emphasis collapsed to 1, e.g. `assault`, `road_runner`, 2) the predicted emphasis runs wild to huge values, e.g. `yars_revenge`, `centipede`. The huge negative predictions are especially detrimental as the gradient directions are flipped.

In Fig. 4 shows mean and median training curves, with standard deviations across 3 seeds for each learning frame and then smoothed *via* a standard 1-D Gaussian filter ($\sigma = 10$) for clarity. Adding auxiliary Monte Carlo loss is a double-edged sword, while improving stability of the time-reversed TD learning, it also brings in higher variance from the Monte Carlo emphasis. For this reason X-ETD(n)-MC exhibited more variance than X-ETD(n) during training, however, it also lead to more stability in score improvement across games, especially mitigating losses where learned emphasis failed to help (see Fig. 3).

What X-ETD(n) approximates is essentially similar to the density ratio between the state distributions of the target and behavior policies in that they both share a backward Bellman equation (Liu et al. 2018; Zhang et al. 2020). Learning density ratios is an active research area but past works usually only tested on benchmarks with low-dimensional observations (e.g. MuJoCo (Todorov, Erez, and Tassa 2012), (Liu et al. 2018; Nachum et al. 2019; Zhang, Liu, and Whiteson 2020; Uehara, Huang, and Jiang 2020; Yang et al. 2020)). In this work, we demonstrate that performance improvement is entirely possible when applying learned emphasis to challenging Atari games using high-dimensional image observations.

Related Work

The idea of learning expected emphatic traces as a function of state has been explored before on the canonical followon trace for backward view TD(λ), to improve trackability of the critic in off-policy actor-critic algorithms (Zhang et al. 2020). However in this work we focus on the n-step trace from Jiang et al. (2021) in the forward view, to improve data efficiency in deep RL. Our proposed stabilization techniques, to facilitate at-scale learning, differ from Zhang et al. (2020). Though Zhang et al. (2020) also use a learned trace to reweight 1-step off-policy TD in the GEM-ETD algorithm, theoretical analyses were not provided. In contrast, we provide a thorough theoretical analysis for X-ETD(n). Finally, we demonstrate the effectiveness of our methods in challenging Atari domains, while Zhang et al. (2020) experiment with only small diagnostic environments.

The idea of bootstrapping in the reverse direction has also been explored by Wang, Bowling, and Schuurmans (2007); Wang et al. (2008); Hallak and Mannor (2017); Gelada and Bellemare (2019) in learning density ratios and by Zhang, Veeriah, and Whiteson (2020) in learning reverse general value functions to represent retrospective knowledge. Besides learning a *scalar* followon trace, van Hasselt et al. (2020) learn a *vector* eligibility trace (Sutton 1988), which, together with Satija, Amortila, and Pineau (2020), inspired our use of an auxiliary Monte Carlo loss.

Several prior works have focused on learning density ratios. These algorithms reweight TD-style updates to the value function by the ratio of the stationary distribution of the target policy to the stationary distribution of the behavior policy (Hallak and Mannor 2017; Liu et al. 2018, 2019; Gelada and Bellemare 2019; Kallus and Uehara 2020). These approaches are inspired by the original approach, called COP-TD (Hallak and Mannor 2017), including a non-linear control algorithm (Gelada and Bellemare 2019). COP-TD is similar to ETD(λ) with state/feature-dependent emphasis (Zhang et al. 2020). In this work, we choose to focus on emphatic weightings building on the highly performant ETD(n) algorithm. ETD(n) achieves state-of-the-art across the Atari suite, whereas the non-linear extension of COP-TD to control (Gelada and Bellemare 2019) only performs well in select games. In the linear prediction setting, COP-TD—even with an additional tuneable step-size parameter—has not been shown to reliably outperform ETD(λ) (Hallak and Mannor 2017), whereas ETD(λ) significantly outperforms classical importance sampling approaches (Ghiassian et al. 2018). A systematic comparison of all these reweighting schemes is currently missing, as well as a careful study of each algorithm's scaling properties. These question are beyond the scope of the current study and are left to future work.

Conclusion

In this paper, we propose a simple time-reversed TD learning algorithm for learning expected emphases that is applicable to non-sequential i.i.d. data. We proved that under certain conditions the resulting algorithm X-ETD(n) has low variance, is stable and convergence to a reasonable fixed point. Furthermore, it improved off-policy learning results upon well-established baselines on Atari 2600 games, demonstrating its generality and wide applicability. In future works, we would like to study X-ETD(n) in more diverse off-policy learning settings using different data sources.

References

Arbuthnot, J. 1712. II. An argument for divine providence, taken from the constant regularity observ'd in the births of both sexes. By Dr. John Arbuthnott, Physitian in Ordinary to Her Majesty, and Fellow of the College of Physitians and the Royal Society. *Philosophical Transactions of the Royal Society of London*, 27(328): 186–190.

Baird, L. 1995. Residual Algorithms: Reinforcement Learning with Function Approximation. *Proceedings of the Twelfth International Conference on Machine Learning*, 30–37.

Bellemare, M. G.; Naddaf, Y.; Veness, J.; and Bowling, M. 2013. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279.

Budden, D.; Hessel, M.; Quan, J.; Kapturowski, S.; Baumli, K.; Bhupatiraju, S.; Guy, A.; and King, M. 2020. RLax: Reinforcement Learning in JAX.

Espeholt, L.; Soyer, H.; Munos, R.; Simonyan, K.; Mnih, V.; Ward, T.; Doron, Y.; Firoiu, V.; Harley, T.; Dunning, I.; Legg, S.; and Kavukcuoglu, K. 2018. IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures. *CoRR*.

Gelada, C.; and Bellemare, M. G. 2019. Off-policy deep reinforcement learning by bootstrapping the covariate shift. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3647–3655.

Ghiassian, S.; Patterson, A.; White, M.; Sutton, R. S.; and White, A. 2018. Online Off-policy Prediction. *CoRR*, abs/1811.02597.

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

Hennigan, T.; Cai, T.; Norman, T.; and Babuschkin, I. 2020. Haiku: Sonnet for JAX.

Hessel, M.; Budden, D.; Viola, F.; Rosca, M.; Sezener, E.; and Hennigan, T. 2020. Optax: composable gradient transformation and optimisation, in JAX!

Hessel, M.; Danihelka, I.; Viola, F.; Guez, A.; Schmitt, S.; Sifre, L.; Weber, T.; Silver, D.; and van Hasselt, H. 2021a. Muesli: Combining Improvements in Policy Optimization. In *International Conference on Machine Learning*. PMLR.

Hessel, M.; Kroiss, M.; Clark, A.; Kemaev, I.; Quan, J.; Keck, T.; Viola, F.; and van Hasselt, H. 2021b. Podracer architectures for scalable Reinforcement Learning.

Hessel, M.; Modayil, J.; van Hasselt, H.; Schaul, T.; Ostrovski, G.; Dabney, W.; Horgan, D.; Piot, B.; Azar, M.; and Silver, D. 2018. Rainbow: Combining Improvements in Deep Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Imani, E.; Graves, E.; and White, M. 2018. An Off-Policy Policy Gradient Theorem Using Emphatic Weightings. *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS 2018)*.

Jiang, R.; Zahavy, T.; White, A.; Xu, Z.; Hessel, M.; Blundell, C.; and van Hasselt, H. 2021. Emphatic Algorithms for Deep Reinforcement Learning. In *International Conference on Machine Learning*. PMLR.

Kallus, N.; and Uehara, M. 2020. Statistically efficient off-policy policy gradients. In *International Conference on Machine Learning*, 5089–5100. PMLR.

Lin, L.-J. 1992. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8(3-4): 293–321.

Liu, Q.; Li, L.; Tang, Z.; and Zhou, D. 2018. Breaking the curse of horizon: Infinite-horizon off-policy estimation. *arXiv preprint arXiv:1810.12429*.

Liu, Y.; Swaminathan, A.; Agarwal, A.; and Brunskill, E. 2019. Off-policy policy gradient with state distribution correction. *arXiv preprint arXiv:1904.08473*.

Mahmood, A. R.; Yu, H.; and Sutton, R. S. 2017. Multi-step off-policy learning without importance sampling ratios. *arXiv preprint arXiv:1702.03006*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Hassabis, D. 2015. Human-level control through deep reinforcement learning. *Nature*.

Nachum, O.; Chow, Y.; Dai, B.; and Li, L. 2019. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *arXiv preprint arXiv:1906.04733*.

Satija, H.; Amortila, P.; and Pineau, J. 2020. Constrained markov decision processes via backward value functions. In *International Conference on Machine Learning*, 8502–8511. PMLR.

Schaul, T.; Quan, J.; Antonoglou, I.; and Silver, D. 2016. Prioritized Experience Replay. In Bengio, Y.; and LeCun, Y., eds., *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

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

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

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

Todorov, E.; Erez, T.; and Tassa, Y. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5026–5033. IEEE.

Uehara, M.; Huang, J.; and Jiang, N. 2020. Minimax weight and q-function learning for off-policy evaluation. In *International Conference on Machine Learning*, 9659–9668. PMLR.

van Hasselt, H.; Doron, Y.; Strub, F.; Hessel, M.; Sonnerat, N.; and Modayil, J. 2018. Deep Reinforcement Learning and the Deadly Triad. *CoRR*, abs/1812.02648.

van Hasselt, H.; Hessel, M.; and Aslanides, J. 2019. When to use parametric models in reinforcement learning? In

Advances in Neural Information Processing Systems 36, NeurIPS.

van Hasselt, H.; Madjiheurem, S.; Hessel, M.; Silver, D.; Barreto, A.; and Borsa, D. 2020. Expected eligibility traces. *arXiv preprint arXiv:2007.01839*.

Wang, T.; Bowling, M.; and Schuurmans, D. 2007. Dual representations for dynamic programming and reinforcement learning. In *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, 44–51. IEEE.

Wang, T.; Bowling, M.; Schuurmans, D.; and Lizotte, D. J. 2008. Stable dual dynamic programming. In *Advances in neural information processing systems*, 1569–1576.

Watkins, C. J. C. H.; and Dayan, P. 2004. Q-learning. *Machine Learning*, 8: 279–292.

Yang, M.; Nachum, O.; Dai, B.; Li, L.; and Schuurmans, D. 2020. Off-policy evaluation via the regularized lagrangian. *arXiv preprint arXiv:2007.03438*.

Zahavy, T.; Xu, Z.; Veeriah, V.; Hessel, M.; Oh, J.; van Hasselt, H.; Silver, D.; and Singh, S. 2020. A Self-Tuning Actor-Critic Algorithm. *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*.

Zhang, S.; Boehmer, W.; and Whiteson, S. 2019. Generalized Off-Policy Actor-Critic. In *Advances in Neural Information Processing Systems*, volume 32.

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

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

Zhang, S.; Veeriah, V.; and Whiteson, S. 2020. Learning Retrospective Knowledge with Reverse Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 33.