Trajectory-Aware Eligibility Traces for Off-Policy Reinforcement Learning

Brett Daley [1 2] **Martha White** [1 2 3] **Christopher Amato** [4] **Marlos C. Machado** [1 2 3]

Abstract

Off-policy learning from multistep returns is crucial for sample-efficient reinforcement learning, but counteracting off-policy bias without exacerbating variance is challenging. Classically, off-policy bias is corrected in a *per-decision* manner: past temporal-difference errors are re-weighted by the instantaneous Importance Sampling (IS) ratio after each action via eligibility traces. Many off-policy algorithms rely on this mechanism, along with differing protocols for *cutting* the IS ratios to combat the variance of the IS estimator. Unfortunately, once a trace has been fully cut, the effect cannot be reversed. This has led to the development of credit-assignment strategies that account for multiple past experiences at a time. These *trajectory-aware* methods have not been extensively analyzed, and their theoretical justification remains uncertain. In this paper, we propose a multistep operator that can express both per-decision and trajectory-aware methods. We prove convergence conditions for our operator in the tabular setting, establishing the first guarantees for several existing methods as well as many new ones. Finally, we introduce Recency-Bounded Importance Sampling (RBIS), which leverages trajectory awareness to perform robustly across λ-values in several off-policy control tasks.

1. Introduction

Reinforcement learning concerns an agent interacting with its environment through trial and error to maximize its expected cumulative reward. One of the great challenges of reinforcement learning is the temporal credit assignment problem (Sutton, 1984): upon receiving a reward, which past actions should be held responsible and, hence, be reinforced? Basic temporal-difference (TD) methods assign credit to the immediately taken action (e.g., Watkins, 1989; Rummery & Niranjan, 1994), bootstrapping from previous experience to learn long-term dependencies. This process requires a large number of repetitions to generate effective behaviors from rewards, motivating research into *multistep* return estimation in which credit is distributed among multiple past actions according to some eligibility rule (e.g., Sutton, 1988).

One challenge of multistep estimators is that they generally have higher variance than 1-step estimators (Kearns & Singh, 2000). This is exacerbated in the off-policy setting, where environment interaction is conducted according to a behavior policy that differs from the target policy for which returns are being estimated. The discrepancy between the two policies manifests mathematically as bias in the return estimation, which can be detrimental to learning if left unaddressed (Precup et al., 2000). Despite these challenges, off-policy learning is important for exploration and sample efficiency. The canonical bias-correction technique is Importance Sampling (IS; Kahn & Harris, 1951), wherein the bias due to the differing policies is eliminated by the product of their probability ratios (Precup et al., 2000). Although IS theoretically resolves the off-policy bias, it can suffer from extreme variance that makes it largely impractical.

Directly managing the variance of the IS estimator has been a fruitful avenue for developing efficient off-policy algorithms. Past work has focused on modifying the individual IS ratios to reduce the variance of the full update: e.g., Tree Backup (Precup et al., 2000), $Q^\pi(\lambda)$ (Harutyunyan et al., 2016), Retrace (Munos et al., 2016), ABQ (Mahmood et al., 2017), and C-trace (Rowland et al., 2020). All of these methods can be implemented online with *per-decision* rules (Precup et al., 2000) that determine how much to reduce, or *cut*, the IS ratio according to the current state-action pair. The re-weighted TD error is then broadcast to previous experiences using eligibility traces (Barto et al., 1983; Sutton, 1984). The decisions made by these algorithms are Markov in the sense that each iterative off-policy correction depends on only the current state-action pair. One issue

[1]Department of Computing Science, University of Alberta, Edmonton, AB, Canada [2]Alberta Machine Intelligence Institute [3]Canada CIFAR AI Chair [4]Khoury College of Computer Sciences, Northeastern University, Boston, MA, USA. Correspondence to: Brett Daley <brett.daley@ualberta.ca>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

with this is that it can lead to suboptimal decisions, since fully cutting a trace cannot be reversed later. In contrast, a *trajectory-aware* method can examine an entire sequence of past state-action pairs to make globally better decisions regarding credit assignment; for example, when a specific transition yields a high IS ratio, a trajectory-aware method can choose to not cut the trace if the product of all previous IS ratios remains small.

Indeed, some existing off-policy methods already conduct offline bias correction in a trajectory-aware manner. Perhaps the simplest example is Truncated IS, where the IS ratio products are pre-calculated offline and then clipped to some finite value (see Section 4). More recently, Munos et al. (2016) suggested a recursive variant of Retrace that automatically relaxes the clipping bound when its historical trace magnitude becomes small; the authors conjectured that this could lead to faster learning. No theoretical analysis has been conducted on trajectory-aware algorithms such as these; their convergence properties are unknown, and the space of possible algorithms has not yet been fully explored.

To better understand these algorithms, and to support new discoveries of efficient algorithms, we introduce a unifying theoretical perspective on per-decision and trajectory-aware off-policy corrections. We propose a multistep operator that accounts for arbitrary dependencies on past experiences, significantly generalizing the per-decision $\mathcal{R}$ operator introduced by Munos et al. (2016). We prove that our operator converges for policy evaluation and control. In the latter case, we remove the assumptions of increasingly greedy policies and pessimistic initialization used by Munos et al. (2016), which has implications for per-decision methods. Finally, we derive a new method from our theory, Recency-Bounded Importance Sampling (RBIS), which performs favorably to other trajectory-aware methods across a wide range of λ-values in an off-policy control task.

2. Preliminaries

We consider Markov Decision Processes (MDPs) of the form $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$. $\mathcal{S}$ and $\mathcal{A}$ are finite sets of states and actions, respectively. Letting $\Delta\mathcal{X}$ denote the set of distributions over a set $\mathcal{X}$, then $P\colon \mathcal{S} \times \mathcal{A} \to \Delta\mathcal{S}$ is the transition function, $R\colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. A policy $\pi\colon \mathcal{S} \to \Delta\mathcal{A}$ determines an agent's probability of selecting a given action in each state. A value function $Q\colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ represents the agent's estimate of the expected return achievable from each state-action pair. For a policy π, we define the operator

$$(P_\pi Q)(s, a) \coloneqq \sum_{s' \in \mathcal{S}} \sum_{a' \in \mathcal{A}} P(s'|s, a)\pi(a'|s')Q(s', a').$$

As a shorthand, we represent value functions and the reward function as vectors in $\mathbb{R}^n$, where $n = |\mathcal{S} \times \mathcal{A}|$. Linear operators such as P_π can hence be interpreted as $n \times n$ square matrices that multiply these vectors, with repeated application corresponding to exponentiation: $P_\pi^t Q = P_\pi(P_\pi^{t-1}Q)$.

In the *policy evaluation* setting, we seek to estimate the expected discounted returns for policy π, given by $Q^\pi \coloneqq \sum_{t=0}^{\infty} \gamma^t P_\pi^t R$. The value function Q^π is the unique fixed point of the Bellman operator $T_\pi Q \coloneqq R + \gamma P_\pi Q$, i.e., it uniquely solves the Bellman equation $T_\pi Q^\pi = Q^\pi$ (Bellman, 1966). In the *control* setting, we seek to estimate the expected returns Q^* under the optimal policy π^*. Q^* is the unique fixed point of the Bellman optimality operator $(TQ)(s, a) \coloneqq \max_\pi (T_\pi Q)(s, a)$, i.e., it uniquely solves the Bellman optimality equation $TQ^* = Q^*$. We are particularly interested in the *off-policy* learning case, where trajectories of the form $(S_0, A_0), (S_1, A_1), (S_2, A_2), \ldots$ are generated by interacting with the MDP using a behavior policy μ, where $\mu \neq \pi$. We define the TD error for policy π at time t as

$$\delta_t^\pi \coloneqq R_t + \gamma \sum_{a' \in \mathcal{A}} \pi(a'|S_{t+1})Q(S_{t+1}, a') - Q(S_t, A_t),$$

where $R_t \coloneqq R(S_t, A_t)$. Let $\rho_k \coloneqq \frac{\pi(A_k|S_k)}{\mu(A_k|S_k)}$ for brevity. Munos et al. (2016) introduced the off-policy operator

$$(\mathcal{R}Q)(s, a) \coloneqq Q(s, a) + \mathbb{E}_\mu\left[\sum_{t=0}^{\infty} \gamma^t \left(\prod_{k=1}^{t} c_k\right) \delta_t^\pi \;\middle|\; (S_0, A_0) = (s, a)\right], \quad (1)$$

where $c_k \coloneqq c(S_k, A_k) \in [0, \rho_k]$. We refer to the product $\prod_{k=1}^{t} c_k$ as the *trace* for (s, a) at time t. If any $c_k < \rho_k$, we say that the trace has been (partially) cut. If any $c_k = 0$, then we have fully cut it. If the trace is fully cut at $t = 1$, i.e., $c_1 = 0$, then $(\mathcal{R}Q)(s, a) = Q(s, a) + \mathbb{E}[\delta_0^\pi \mid (S_0, A_0) = (s, a)] = R(s, a) + \gamma\mathbb{E}[\sum_{a' \in \mathcal{A}} \pi(a'|S_1)Q(S_1, a')|(S_0, A_0) = (s, a)]$, which is the standard 1-step bootstrap target like in TD(0) (Sutton, 1988). Notice that each c_k is Markov, as it depends only on (S_k, A_k) and is otherwise independent of the preceding trajectory. In other words, the update for $\mathcal{R}$ can be calculated *per decision* (Precup et al., 2000), permitting an efficient online implementation with eligibility traces.

3. Trajectory-Aware Eligibility Traces

While per-decision traces are convenient from a computational perspective, they require making choices about how much to cut the trace without considering the effects of previous choices. This can lead to suboptimal decisions; for example, if the trace is cut by setting $c_k = 0$ at some timestep, then the effect cannot be reversed later. Regardless of whatever new experiences are encountered by the agent, experiences before time k will be ineligible for credit assignment, resulting in an opportunity cost. In fact, this exact

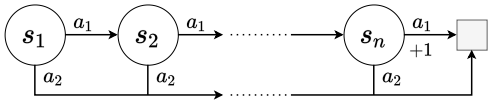

Figure 1. The Tightrope Problem. Starting from state s_1, the agent must take a specific sequence of n actions to receive $+1$ reward.

phenomenon is why Watkins' Q(λ) (Watkins, 1989) often learns more slowly than Peng's Q(λ) (Peng & Williams, 1996), even though the former avoids off-policy bias (Sutton & Barto, 1998; Daley & Amato, 2019; Kozuno et al., 2021). The same effect (but to a lesser extent) impacts Tree Backup and Retrace, where $c_k \leq 1$ always in Eq. (1), implying that the traces for past experiences can never increase.

We illustrate this phenomenon in a small, deterministic MDP that we call the Tightrope Problem (see Figure 1). The environment consists of n sequential, non-terminal states with two actions a_1, a_2 available. The agent starts in state s_1 and advances from s_i to s_{i+1} whenever it takes action a_1. If $i = n$, then the episode terminates and the agent receives $+1$ reward. Taking action a_2 in any state immediately terminates the episode with no reward. Clearly, the optimal policy is to execute a_1 regardless of the state.

Now consider the following off-policy learning scenario. Suppose the agent's behavior policy μ is uniform random, but the target policy π is ϵ-greedy with respect to a value function Q. For each state s, it follows that $\pi(a|s) = 1 - \epsilon$ if $a = \mathrm{argmax}_{a'} Q(s, a')$ and $\pi(a|s) = \epsilon$ otherwise. We assume ϵ is small in the sense that $\epsilon < \frac{1}{2}$, and that $\gamma = 1$. Suppose now that the agent successfully receives the $+1$ reward during an episode, implying that it took action a_1 on every timestep. We can compute the eligibility of the initial state-action pair (s_1, a_1) as an expression in the number k of incorrect actions in the greedy policy (i.e., where $\mathrm{argmax}_{a'} Q(s, a') \neq a_1$). Letting $\lambda \in [0, 1]$ be a decay parameter, the standard IS estimator (which does not cut traces when $\lambda = 1$) provides an eligibility of

$$\left(\lambda \frac{1-\epsilon}{1/2}\right)^{n-k} \left(\lambda \frac{\epsilon}{1/2}\right)^{k} = \lambda^n [2(1-\epsilon)]^{n-k} (2\epsilon)^k. \quad (2)$$

This value can be greater than 1 when $k \ll n$, which suggests that the agent's behavior should be heavily reinforced when the greedy policy agrees closely with the optimal policy; however, a per-decision method like Retrace, which cuts traces without considering the full trajectory (see Section 4), ultimately assigns a much lower eligibility:

$$\left[\lambda \min\left(1, \frac{1-\epsilon}{1/2}\right)\right]^{n-k} \left[\lambda \min\left(1, \frac{\epsilon}{1/2}\right)\right]^{k} = \lambda^n (2\epsilon)^k.$$

The eligibility now decays monotonically for every suboptimal action in the greedy policy, illuminating how per-decision trace cutting can lead to excessively small eligibilities, especially when ϵ is close to 0.

This issue stems from the fact that Retrace is not *aware* of its past eligibilities, and continues to decay them even when they already form an underestimate compared to IS. This issue is not unique to Retrace, and affects other per-decision methods like Tree Backup. Instead, a *trajectory-aware* method that can actively adapt its trace-cutting behavior based on the magnitude of past eligibilities would be better.

One way to obtain a trajectory-aware method is to compute the exact IS product in Eq. (2), and then make adjustments to it to achieve certain properties (e.g., convergence and variance reduction). For example, Truncated IS (see Section 4) simply imposes a fixed bound on the IS estimator:

$$\lambda^n \min\big(1,\ [2(1-\epsilon)]^{n-k} (2\epsilon)^k\big)\,. \quad (3)$$

Ignoring λ, Truncated IS reduces the eligibility only when it exceeds a pre-specified threshold, effectively avoiding trace cuts when the true IS estimate is small. In Section 6, we propose an algorithm, RBIS, which achieves a similar effect using a recursive, time-decaying threshold.

As this example demonstrates, it can be advantageous to consider the agent's past experiences to produce better decisions regarding credit assignment. One of our principal contributions is the proposal and analysis of an off-policy operator $\mathcal{M}$ that encompasses this possibility. Let $\mathcal{F}_t := (S_0, A_0), (S_1, A_1), \ldots, (S_t, A_t)$. We define $\mathcal{M}$ such that

$$(\mathcal{M}Q)(s,a) := Q(s,a) + \mathbb{E}_\mu\left[\sum_{t=0}^{\infty} \gamma^t \beta_t \delta_t^\pi \;\middle|\; (S_0, A_0) = (s, a)\right], \quad (4)$$

where $\beta_t := \beta(\mathcal{F}_t)$ is a trace that generally depends on the history $\mathcal{F}_t$. We define $\beta_0 := 1$ to ensure that the first TD error, δ_0^π, is applied. In Section 5, we characterize the values of β_t for $t \geq 1$ that lead to convergence.

The major analytical challenge of $\mathcal{M}$—and its main novelty—is the complex dependence on the sequence $\mathcal{F}_t$. This makes the operator difficult to analyze mathematically, as the terms in the series $1 + \gamma\beta_1 + \gamma^2\beta_2 + \cdots$ generally share no common factors that would allow a recursive formula for eligibility traces. Some off-policy methods, however, cannot be described by factored traces, and therefore removing this assumption is necessary to understand existing algorithms (see Section 4), while also paving the way for new credit-assignment methods. In the special case where β_t does factor into Markov coefficients, i.e., $\beta_t = \prod_{k=1}^{t} c_k$, then Eq. (4) reduces to Eq. (1), taking us back to the per-decision setting studied by Munos et al. (2016). *$\mathcal{M}$, therefore, unifies per-decision and trajectory-aware methods.*

4. Unifying Off-Policy Algorithms

The operator $\mathcal{M}$ is a strict generalization of the previous operator considered for trace-based methods, allowing us to express existing algorithms in this form. We provide a non-exhaustive list of examples below with the corresponding β_t used in $\mathcal{M}$. For brevity, let $\Pi_t := \prod_{k=1}^{t} \rho_k$.

Importance Sampling: $\beta_t = \lambda^t \Pi_t$ (Kahn & Harris, 1951). The standard approach for correcting off-policy bias. Although it is the only unbiased estimator in this list (if $\lambda = 1$), it suffers from high variance, making it difficult to utilize.

$\mathbf{Q}^\pi(\lambda)$: $\beta_t = \lambda^t$ (Harutyunyan et al., 2016). A straightforward algorithm that decays the TD errors by a fixed constant. The algorithm does not require explicitly knowing μ, which is desirable, but can diverge if π and μ differ too much (Harutyunyan et al., 2016, Theorem 1).

Tree Backup: $\beta_t = \prod_{k=1}^{t} \lambda \pi(A_k|S_k)$ (Precup et al., 2000). A method that automatically cuts traces according to the product of probabilities under π, which forms a conservative lower bound on the IS estimate. Tree Backup converges for any behavior policy μ, but it is not efficient since traces are cut excessively—especially in the on-policy case.

Retrace: $\beta_t = \prod_{k=1}^{t} \lambda \min(1,\ \rho_k)$ (Munos et al., 2016). A convergent algorithm for arbitrary policies π and μ that remains efficient in the on-policy case because it does not cut traces (if $\lambda = 1$); however, the fact that β_t never increases can cause the trace products to decay too quickly in practice (Mahmood et al., 2017; Rowland et al., 2020).

All of the above can be analyzed using a per-decision operator. The next two, on the other hand, have weightings based on the entire trajectory. We use the theory for our general $\mathcal{M}$ operator to prove properties about these methods.

Recursive Retrace: $\beta_t = \lambda \min(1, \beta_{t-1}\rho_t)$ (Munos et al., 2016). A modification to Retrace conjectured to lead to faster learning. It clips large products of ratios, rather than individual ratios. Its convergence for control is an open question, which we solve in Section 5.

Truncated Importance Sampling: $\beta_t = \lambda^t \min(1,\ \Pi_t)$ (Ionides, 2008). A simple but effective method to combat the variance of IS. Variations of this algorithm have been applied in the reinforcement learning literature (e.g., Uchibe & Doya, 2004; Wawrzyński & Pacut, 2007; Wawrzyński, 2009; Wang et al., 2017), but, to our knowledge, its convergence in an MDP setting has not been studied. In Section 5.3, we show that it can diverge in at least one off-policy problem.

5. Convergence Analysis

In this section, we study the convergence properties of the $\mathcal{M}$ operator for policy evaluation and control. It will be convenient to re-express Eq. (4) in vector notation for our analysis. To do this, let us first bring the expectation inside the sum, by linearity of expectation:

$$(\mathcal{M}Q)(s,a) = Q(s,a) + \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_\mu\big[\beta_t \delta_t^\pi \;\big|\; (S_0, A_0) = (s,a)\big]. \tag{5}$$

To write Eq. (5) in vector form, we define an operator B_t such that, for an arbitrary vector X in $\mathbb{R}^n$,

$$(B_t X)(s,a) := \mathbb{E}_\mu\big[\beta_t X(S_t, A_t) \;\big|\; (S_0, A_0) = (s,a)\big], \tag{6}$$

allowing us to express the $\mathcal{M}$ operator as

$$\mathcal{M}Q = Q + \sum_{t=0}^{\infty} \gamma^t B_t (T_\pi Q - Q). \tag{7}$$

B_t is a linear operator and hence can be represented as a matrix in $\mathbb{R}^{n\times n}$, the elements of which are nonnegative. Each element of B_t, row-indexed by (s,a) and column-indexed by (s',a'), has the form

$$B_t((s,a),(s',a')) = \Pr_\mu((S_t,A_t) = (s',a') \,|\, (S_0,A_0) = (s,a)) \times\ \mathbb{E}_\mu\big[\beta_t \,\big|\, (S_0,A_0) = (s,a), (S_t,A_t) = (s',a')\big]. \tag{8}$$

We justify this form in Appendix A. Note that $B_0 = I$, the identity matrix, because of our earlier definition of $\beta_0 := 1$. In the following sections, all inequalities involving vectors or matrices should be interpreted element wise. We let $\|X\| := \|X\|_\infty$ for a matrix (or vector) X, which corresponds to the maximum absolute row sum of X. We also define $\mathbf{1} \in \mathbb{R}^n$ to be the vector of ones, such that $X\mathbf{1}$ gives the row sums of X.

5.1. Convergence for Policy Evaluation

We start in the off-policy policy evaluation setting. Specifically, our goal is to prove that the repeated application of the $\mathcal{M}$ operator to an arbitrarily initialized vector $Q \in \mathbb{R}^n$ converges to Q^π.

Condition 5.1. $\beta_t \le \beta_{t-1}\rho_t,\ \forall\, \mathcal{F}_t,\ \forall\, t \ge 1.$

Theorem 5.2. *If Condition 5.1 holds, then $\mathcal{M}$ is a contraction mapping with Q^π as its unique fixed point. Consequently, $\lim_{i\to\infty} \mathcal{M}^i Q = Q^\pi,\ \forall\, Q \in \mathbb{R}^n$.*

Proof. In Lemma B.1 (Appendix B.1), we show that Q^π is a fixed point of $\mathcal{M}$ and that

$$\mathcal{M}Q - Q^\pi = Z(Q - Q^\pi), \tag{9}$$

where $Z := \sum_{t=1}^{\infty} \gamma^t (B_{t-1}P_\pi - B_t)$. In Lemma B.2 (Appendix B.2), we also show that $Z \ge 0$ and $Z\mathbf{1} \le \gamma$ using the assumption that $\beta_t \le \beta_{t-1}\rho_t, \forall\, \mathcal{F}_t, \forall\, t \ge 1$ (Condition 5.1). Consequently, $Z(Q - Q^\pi)$ is a vector whose components

each comprise a nonnegative-weighted combination of the components of $Q - Q^\pi$, where the weights add up to at most γ. This means $\|\mathcal{M}Q - Q^\pi\| \leq \gamma\|Q - Q^\pi\|$, and $\mathcal{M}$ is a contraction mapping. Its fixed point, Q^π, must therefore be unique by the Banach fixed-point theorem, implying that $\lim_{i\to\infty} \mathcal{M}^i Q = Q^\pi$ for every $Q \in \mathbb{R}^n$ when $\gamma < 1$. □

Given that the ratio $\frac{\beta_t}{\beta_{t-1}}$ is bounded by ρ_t (Condition 5.1), the $\mathcal{M}$ operator converges to Q^π. Intuitively, we can think of this ratio as the *effective* per-decision factor at time t; convergence is guaranteed whenever this factor is no greater than ρ_t, analogous to the convergence result for the $\mathcal{R}$ operator (Munos et al., 2016, Theorem 1). Our theorem implies the existence of a space of convergent trajectory-aware algorithms, because each trace β_t can be chosen arbitrarily so long as it always satisfies the bound on this ratio.

5.2. Convergence for Control

We now consider the more challenging setting of control. Given sequences of target policies $(\pi_i)_{i\geq 0}$ and behavior policies $(\mu_i)_{i\geq 0}$, we aim to show that the sequence of value functions $(Q_i)_{i\geq 0}$ given by $Q_{i+1} \coloneqq \mathcal{M}_i Q_i$ converges to Q^*. Here, $\mathcal{M}_i$ is the $\mathcal{M}$ operator defined for π_i and μ_i.

Compared to the convergence proof of the $\mathcal{R}$ operator (Munos et al., 2016, Theorem 2), the main novelty of our proof is the fact that the traces under $\mathcal{M}$ are not Markov. Consequently, we require new techniques to establish bounds on $Q - Q^*$, since Eq. (4) is not representable as an infinite geometric series and so the summation does not have a closed-form expression. We additionally relax two assumptions in the previous work, on initialization of the value function and on increasing greediness of the policy. We require only that the target policies become *greedy in the limit*. We say that a sequence of policies is greedy in the limit if $T_{\pi_i} Q_i \to TQ_i$ as $i \to \infty$. We discuss the significance of these relaxations to the assumptions in Section 5.4.

First, let $C_i \coloneqq \sum_{t=0}^{\infty} \gamma^t B_t$ for the policies π_i and μ_i, and write the $\mathcal{M}$ operator at iteration i as

$$\mathcal{M}_i Q = Q + C_i(T_{\pi_i} Q - Q). \tag{10}$$

We now present our convergence theorem for control.

Theorem 5.3. *Consider a sequence of target policies $(\pi_i)_{i\geq 0}$ and a sequence of arbitrary behavior policies $(\mu_i)_{i\geq 0}$. Let Q_0 be an arbitrary vector in $\mathbb{R}^n$ and define the sequence $Q_{i+1} \coloneqq \mathcal{M}_i Q_i$, where $\mathcal{M}_i$ is the operator defined by Eq. (10). Assume that $(\pi_i)_{i\geq 0}$ is greedy in the limit, and let $\epsilon_i \geq 0$ be the smallest constant such that $T_{\pi_i} Q_i \geq TQ_i - \epsilon_i \|Q_i\| \mathbf{1}$. If Condition 5.1 holds for all i, then*

$$\|\mathcal{M}_i Q_i - Q^*\| \leq \gamma \|Q_i - Q^*\| + \frac{\epsilon_i}{1-\gamma}\|Q_i\|, \tag{11}$$

and, consequently, $\lim_{i\to\infty} Q_i = Q^*$.

Proof (sketch; full proof in Appendix B.3). We define matrices Z_i and Z_i^*, which correspond to Z in Eq. (9) for target policies π_i and π^*, respectively, and behavior policy μ_i. We then derive the inequalities

$$Z_i^*(Q_i - Q^*) - \epsilon_i \|Q_i\| C_i \mathbf{1} \leq \mathcal{M}_i Q_i - Q^* \leq Z_i(Q_i - Q^*),$$

which together imply Eq. (11). Thus, $\mathcal{M}_i$ is nearly a contraction mapping with Q^* as its unique fixed point, excepting the influence of the $O(\|Q_i\|)$ term. However, the greedy-in-the-limit target policies guarantee that $\epsilon_i \to 0$. Showing that $\|Q_i\|$ remains finite completes the proof because $\|Q_i - Q^*\| \to 0$ must follow. □

The convergence criteria for β_t (Condition 5.1) is the same for both policy evaluation and control. In fact, the only additional assumption we need for control is the greedy-in-the-limit target policies. Crucially, the proof allows arbitrary behavior policies and an arbitrary value function initialization Q_0, which we further discuss in Section 5.4.

5.3. Examples of Convergence and Divergence

The generality of the $\mathcal{M}$ operator means that it provides convergence guarantees for a number of credit-assignment methods that we did not discuss in Section 4. These include variable or past-dependent λ-values (e.g., Watkins, 1989; Singh & Sutton, 1996; Yu et al., 2018). All of these can be represented in a common form and shown to satisfy Condition 5.1; convergence for policy evaluation and control for the instantiated trajectory-aware operator follows as a corollary, since Condition 5.1 is sufficient to apply Theorems 5.2 and 5.3.

Proposition 5.4. *Any traces expressible in the form $\beta_t = \prod_{k=1}^{t} \lambda(\mathcal{F}_k)\rho_k$, $\lambda(\mathcal{F}_k) \in [0, 1]$, satisfy Condition 5.1.*

Proof. $\beta_t = \beta_{t-1}\lambda(\mathcal{F}_t)\rho_t \leq \beta_{t-1}\rho_t$. □

In Section 4, we also discussed two existing trajectory-aware methods whose convergence is unknown. We show that Recursive Retrace satisfies our required condition.

Proposition 5.5. *Recursive Retrace satisfies Condition 5.1.*

Proof. For Recursive Retrace, $\beta_t = c_t \beta_{t-1}$, where $c_t = \lambda \min\left(\frac{1}{\beta_{t-1}},\ \rho_t\right)$ (Munos et al., 2016, Eq. 9). This means

$$\begin{aligned}\beta_t &= \lambda \min\left(\frac{1}{\beta_{t-1}},\ \rho_t\right)\beta_{t-1} \\ &= \lambda \min\left(1,\ \beta_{t-1}\rho_t\right) \\ &\leq \beta_{t-1}\rho_t,\end{aligned} \tag{12}$$

which is the bound required by Condition 5.1. □

Unfortunately, the traces for Truncated IS do not always satisfy the required bound.

Proposition 5.6. *Truncated IS may violate Condition 5.1.*

Proof. We show this by providing a counterexample. Recall that Truncated IS has $\beta_t = \lambda^t \min(1,\ \Pi_t)$. Assume a trajectory $\mathcal{F}_t$ such that $\Pi_{t-1} = 2$ and $\frac{1}{2} < \rho_t < \lambda$. (It is straightforward to define an MDP, behavior policy, and target policy to create such a trajectory.) Because $\rho_t > 1/\Pi_{t-1}$, then $\Pi_t = \Pi_{t-1}\rho_t > 1$. Thus, $\beta_t = \lambda^t$ and $\beta_{t-1} = \lambda^{t-1}$, and Condition 5.1 is violated because $\frac{\beta_t}{\beta_{t-1}} = \lambda \not\leq \rho_t$. □

Because Theorems 5.2 and 5.3 cannot be applied, the precise conditions under which Truncated IS converges remains an open problem. We do know Condition 5.1 is sufficient for convergence, but it is unlikely to be strictly necessary. This is because our proofs of Theorems 5.2 and 5.3 use this assumption to guarantee that the matrix Z in Eq. (9) has nonnegative elements, making it straightforward to show that its row sums are sufficiently bounded to guarantee that $\mathcal{M}$ is a contraction mapping. However, $\mathcal{M}$ could remain a contraction mapping even when Z has negative elements, so long as $\|Z\| < 1$. It could theoretically be the case for Truncated IS that Z occasionally contains negative elements but $\|Z\|$ is still bounded enough to permit convergence.

Nevertheless, we are able to find at least one off-policy problem for which this is not true, implying that certain initializations of the value function could ultimately cause Truncated IS to diverge.

Counterexample 5.7 (Off-Policy Truncated IS)**.** *Consider Truncated IS with $\lambda = 1$, so $\beta_t = \min(1,\ \Pi_t)$. Assume the MDP has one state and two actions: $\mathcal{S} = \{s\}$ and $\mathcal{A} = \{a_1, a_2\}$, the behavior policy μ is uniform random, and π selects a_1 with probability $p \in (0, 1)$ and selects a_2 otherwise. When $p = 0.6$ and $\gamma = 0.94$, then $\|Z\| > 1$.*

Many choices of p and γ make $\|Z\| > 1$, but we discuss specific ones for the counterexample in Appendix C.1.

Compared to the per-decision case, where Munos et al. (2016) showed that arbitrary trace cuts always produce a convergent algorithm, this result is surprising. Why would the analogous result—in which we ensure that $\beta_t \leq \Pi_t$ for all timesteps—not hold here? After all, clipping β_t such that it never exceeds the IS estimate Π_t would be expected to simply incur bias in the return estimation. For some insight, assume the following expectations are conditioned on $(S_0, A_0) = (s, a)$, and observe that Eq. (4) is equivalent to

$$\mathbb{E}_\mu\left[\sum_{t=0}^{\infty} \gamma^t \beta_t R_t\right] + \mathbb{E}_\mu\left[\sum_{t=1}^{\infty} \gamma^t (\beta_{t-1}\rho_t - \beta_t) Q(S_t, A_t)\right].$$

We show the derivation in Appendix A. The first term is a (partially) bias-corrected estimate of the discounted return. The second term is a weighted combination of value-function bootstraps, whose weights are nonnegative when Condition 5.1 is met. If the condition is violated on any timestep, then we may actually be subtracting bootstraps from the return estimate, which does not seem sensible. We believe this is related to the root cause of divergence in Counterexample 5.7; however, it remains open whether Condition 5.1 is necessary or merely sufficient.

As our next counterexample example will demonstrate, this effect can even cause divergence in *on-policy* settings.

Counterexample 5.8 (On-Policy Binary Traces)**.** *Assume the MDP has one state and two actions: $\mathcal{S} = \{s\}$ and $\mathcal{A} = \{a_1, a_2\}$. Define a trajectory-aware method such that $\beta_t = 1$ if $A_t = a_1$ and $\beta_t = 0$ if $A_t = a_2$ (without loss of generality). Assume π and μ are uniform random. When $\gamma \geq \frac{2}{3}$, then $\|Z\| \geq 1$.*

We provide details in Appendix C.2. Even though $\beta_t \leq \Pi_t = 1$ always, we are able to produce a non-contraction. The method either fully cuts a trace or does not cut it at all, producing backups that consist of a sparse sum of on-policy TD errors. It is therefore surprising that divergence occurs. For the same reason we described above, the non-Markov nature of the trace appears to sometimes cause adverse bootstrapping effects; in this instance, the ability to examine each trajectory allows the method to strategically de-emphasize certain state-action pairs, ultimately producing a detrimental effect on learning. Notice that Condition 5.1 is indeed violated in this case because there is always some chance that $\beta_t = 1$ after $\beta_{t-1} = 0$. If we add the restriction that $\beta_{t-1} = 0 \implies \beta_t = 0$, i.e., we permanently cut the traces, then convergence is reestablished by Theorem 5.2.

5.4. Discussion

In this section, we summarize our main theoretical contributions and their significance. We focused on characterizing the contraction properties of the $\mathcal{M}$ operator, both for policy evaluation and control, in the tabular setting. These results parallel those for the $\mathcal{R}$ operator underlying Retrace, where $\mathcal{M}$ is a strict generalization of $\mathcal{R}$. These results indicate that using fixed-point updates, like dynamic programming and temporal difference learning updates, may have divergence issues. It does not, however, imply other algorithms, such as gradient-based algorithms, cannot find these fixed points. We show the fixed points still exist and are unbiased, but that algorithms based on iterating with the $\mathcal{M}$ operator might diverge.

Removal of the Markov assumption. Removing the Markov (per-decision) assumption of the $\mathcal{R}$ operator (Munos et al., 2016) to enable trajectory-aware eligibility traces was our primary goal. When the trace factors are Markov, the operator B_t is independent of t, allowing the sum $\sum_{t=0}^{\infty} \gamma^t B_t$ to be reduced to $\sum_{t=0}^{\infty} (\gamma P_{c\mu})^t$ for a linear operator $P_{c\mu}$. The resulting geometric series can then be evaluated analytically, as was done by Munos et al. (2016). In our proofs, we avoided the Markov assumption by directly analyzing the infinite summation, which generally

does not have a closed-form expression. Our work is the first to do this, establishing the first convergence guarantees for general trajectory-aware methods.

Arbitrary initialization of the value function. We permit any initialization of Q_0 in the control setting. In contrast, Munos et al. (2016) made the assumption that $T_{\pi_0} Q_0 - Q_0 \geq 0$ in order to produce a lower bound on $\mathcal{R}_i Q_i - Q^*$, accomplished in practice by a pessimistic initialization of the value function: $Q_0(s,a) = -\|R\| \,/\, (1-\gamma), \forall\, (s,a) \in \mathcal{S} \times \mathcal{A}$. Since $\mathcal{R}$ is a special case of our operator $\mathcal{M}$ where each trace β_t factors into Markov coefficients, we deduce as a corollary that Retrace and all other algorithms described by $\mathcal{R}$ do not require pessimistic initialization for convergence.

Greedy-in-the-limit policies. Our requirement of greedy-in-the-limit target policies in Theorem 5.3 is less restrictive than the increasingly greedy policies proposed by Munos et al. (2016). We need only $\lim_{i\to\infty} T_{\pi_i} Q_i = TQ_i$, and we do not force the sequence of target policies to satisfy $P_{\pi_{i+1}} Q_{i+1} \geq P_{\pi_i} Q_{i+1}$. This implies that the agent may target non-greedy policies for any finite period of time, as long as the policies do eventually become arbitrarily close to the greedy policy. As a corollary, increasingly greedy policies are not necessary for the optimal convergence of Retrace and other per-decision methods.

6. Recency-Bounded Importance Sampling

Theorem 5.2 guarantees convergence to Q^π whenever Condition 5.1 holds, but we do not expect that all choices of coefficients that satisfy this condition will perform well in practice. At one extreme, if $\beta_t \leq \prod_{k=1}^{t} \lambda \min(1,\ \rho_k)$ for every $\mathcal{F}_t$, then we have a method that cuts coefficients more aggressively than Retrace does; it seems unlikely that such a method would learn faster than Retrace, or other per-decision methods. At the other extreme, when $\beta_t = \Pi_t$ for every $\mathcal{F}_t$, we recover the standard IS estimator, which suffers from high variance and is often ineffectual. We therefore know that it is possible to have a method that preserves traces *too much*, to the point of being detrimental. Thus, it is important to maintain some minimum efficiency by avoiding unnecessary cuts, yet equally important to control the overall variance of the traces.

Intuitively, we want something that falls between Retrace and IS in terms of trace cutting, in order to quickly backpropagate credit while still managing the variance. We further hypothesize that effective trajectory-aware methods will first compute $\beta_{t-1}\rho_t$—i.e., the maximum trace permitted by Condition 5.1—and then apply some transformation that limits its magnitude to reduce variance. This ensures that traces are cut only as needed.

We propose one method, Recency-Bounded Importance Sampling (RBIS), which achieves this by cutting the traces only when they exceed an exponentially decaying threshold. Specifically, we define

$$\beta_t = \min(\lambda^t,\ \beta_{t-1}\rho_t). \tag{RBIS}$$

It is easy to see that RBIS always converges, by construction.

Proposition 6.1. *RBIS satisfies Condition 5.1.*

Proof. $\beta_t = \min(\lambda^t,\ \beta_{t-1}\rho_t) \leq \beta_{t-1}\rho_t.$ □

For further insight, we unroll the recursion to obtain

$$\begin{aligned}
&\min(\lambda^t,\ \beta_{t-1}\rho_t) \\
&\quad = \min(\lambda^t,\ \min(\lambda^{t-1},\ \beta_{t-2}\rho_{t-1})\rho_t) \\
&\quad \cdots \\
&\quad = \min(\lambda^t,\ \lambda^{t-1}\rho_t,\ \lambda^{t-2}\rho_{t-1}\rho_t,\ \ldots,\ \Pi_t). \qquad (13)
\end{aligned}$$

RBIS effectively takes the minimum of all past, discounted n-step IS estimates. This reveals another property of RBIS: its traces are never less than those of Retrace, because

$$\begin{aligned}
\prod_{k=1}^{t} \lambda \min(1,\ \rho_k) &\leq \prod_{k=1}^{t} \min(\lambda,\ \rho_k) \\
&\leq \lambda^{t-j} \prod_{k=t-j+1}^{t} \rho_k,\ \forall\, j \in \{0, 1, \ldots, t\}.
\end{aligned}$$

Since the inequality is true for all j, it is not possible for Retrace's traces to exceed any of the arguments to the $\min$ function in Eq. (13). We have achieved exactly what we wanted earlier: a method that falls somewhere between Retrace and IS in regard to trace cutting. This does not automatically mean that RBIS will outperform Retrace, though, since preserving the magnitude of the trace β_t too much can lead to high variance. However, we do expect RBIS to perform well in decision-making problems in which a few critical actions largely determine the long-term outcome of an episode. In such scenarios, the agent's bottleneck to learning is its ability to assign meaningful credit to these critical actions over a potentially long time horizon.

In order to test this empirically, we construct an environment called the Bifurcated Gridworld (see Figure 2). This 5×5 deterministic gridworld has walls arranged such that two unequal-length paths from the start (S) to the goal (G) are available. The agent may move up, down, left, or right; taking any of these actions in the goal yields a reward of $+1$ and terminates the episode. The problem is discounted ($\gamma = 0.9$) to encourage the agent to learn the shorter path. Importantly, the action taken at the bifurcation (B) solely determines which path the agent follows, and quickly assigning credit to this state is paramount to learning the task.

We compare RBIS against Retrace, Truncated IS, and Recursive Retrace when learning this task from off-policy

data. Both behavior and target policies were ϵ-greedy with respect to the value function Q. The target policy used $\epsilon = 0.1$. The behavior policy used a piecewise schedule: $\epsilon = 1$ for the first 5 episodes and then $\epsilon = 0.2$ afterwards. The agents learned from online TD updates with eligibility traces (see Appendix D for pseudocode). The policies were updated only at the end of each episode, and then the discounted return obtained by a near-greedy policy ($\epsilon = 0.05$) was evaluated. The area under the curve (AUC) of each resulting learning curve was calculated, with the highest AUC achieved over a grid search of stepsizes being plotted for each λ-value in Figure 2. We averaged the results over 1,000 independent trials and indicate the 95% confidence interval by the shaded regions. In Appendix E, we repeated the experiment for three more gridworld topologies; the obtained results are qualitatively similar to Figure 2. Our experiment code is available online.[1]

We make several observations regarding the results in Figure 2. First, the peak performance obtained by RBIS is significantly higher than that of the other three methods. This is notable because both Truncated IS and Recursive Retrace are also trajectory aware, indicating that different implementations of trajectory awareness are beneficial to varying degrees. In particular, the preservation of long-term eligibilities is not sufficient on its own to guarantee strong performance in general, as it appears that *when* and *how much* the traces are cut are important considerations as well. The role of λ as a decay hyperparameter is evidently critical for all methods to achieve their maximum performance, since $\lambda = 1$ never leads to the fastest learning. In fact, $\lambda \to 1$ is especially catastrophic for Truncated IS, which we believe is related to the divergence issue identified in Section 5.3. Finally, Retrace degrades less for larger λ, likely because it cuts traces more. It would be interesting to develop a trajectory-aware method that obtains the robustness of RBIS but also accounts for larger λ-values.

7. Conclusion

In this work, we extended theory for per-decision eligibility traces to trajectory-aware traces. This extension allows us to consider a broader family of algorithms, with more flexibility in obtaining off-policy corrections. Specifically, we introduced the $\mathcal{M}$ operator as a generalization of the $\mathcal{R}$ operator, and a sufficient condition to ensure convergence under $\mathcal{M}$. Using our general result, we established the first convergence guarantee for an existing trajectory-aware method, Recursive Retrace, in the control setting. We also showed that Truncated IS may violate our condition and provided a counterexample showing that it can diverge.

[1] https://github.com/brett-daley/trajectory-aware-etraces

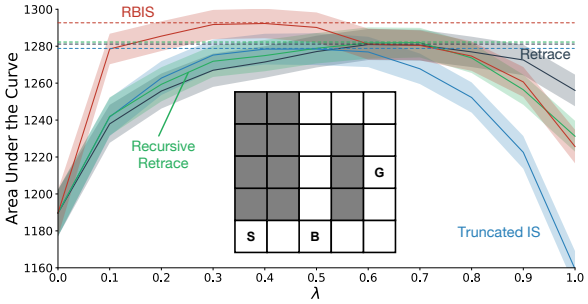

Figure 2. The Bifurcated Gridworld environment. The choice made at B greatly impacts the discounted return ultimately earned. We plot the AUC obtained by four off-policy methods across the λ-spectrum. The dashed horizontal lines mark the highest AUC achieved by each method.

We also proposed a new trajectory-aware method, RBIS, that demonstrates one instance of how trajectory awareness can be utilized for faster learning in off-policy control tasks. RBIS is able to outperform the other trajectory-aware methods that we tested in the Bifurcated Gridworld, suggesting that it possesses at least one unique property that is beneficial for long-term, off-policy credit assignment. It would be interesting to search for additional beneficial properties in future work, in order to better characterize off-policy methods that reliably lead to efficient and stable learning in challenging reinforcement learning environments.

This work focused on convergence *in expectation*; a natural next step is to extend this result to the stochastic algorithms used in practice. Previous results for TD learning rely primarily on the properties of the expected update, with additional conditions on the noise in the update and appropriately annealed stepsizes (see Bertsekas & Tsitsiklis, 1996, Section 4.3). Similar analysis should be applicable, given that we know the expected update with $\mathcal{M}$ is a contraction mapping when Condition 5.1 is met.

An important next step is extending these methods and results to function approximation. Incorporating these traces into deep reinforcement learning methods that rely on experience replay (Lin, 1992) should be straightforward. Multistep returns can be computed offline in the replay memory, and then randomly sampled in minibatches to train the neural network. Using a TD learning update, though, can suffer from convergence issues under function approximation and off-policy learning; this has been previously resolved by developing gradient-based updates (Sutton et al., 2009; Touati et al., 2018). An important next step is to develop a gradient-based trajectory-aware algorithm. The contraction properties of the operator still impact the quality of the solution, as has been shown to be the case for other off-policy approaches (Patterson et al., 2022). The insights in this work, therefore, may provide insights on how to get quality solutions with gradient-based approaches.

References

Barto, A. G., Sutton, R. S., and Anderson, C. W. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 834–846, 1983.

Bellman, R. Dynamic programming. *Science*, 153(3731): 34–37, 1966.

Bertsekas, D. P. and Tsitsiklis, J. N. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

Daley, B. and Amato, C. Reconciling λ-returns with experience replay. In *Advances in Neural Information Processing Systems*, pp. 1133–1142, 2019.

Harutyunyan, A., Bellemare, M. G., Stepleton, T., and Munos, R. Q(λ) with off-policy corrections. In *International Conference on Algorithmic Learning Theory*, pp. 305–320, 2016.

Ionides, E. L. Truncated importance sampling. *Journal of Computational and Graphical Statistics*, 17(2):295–311, 2008.

Kahn, H. and Harris, T. E. Estimation of particle transmission by random sampling. *National Bureau of Standards: Applied Mathematics Series*, 12:27–30, 1951.

Kearns, M. J. and Singh, S. P. Bias-variance error bounds for temporal difference updates. In *Conference on Learning Theory*, pp. 142–147, 2000.

Kozuno, T., Tang, Y., Rowland, M., Munos, R., Kapturowski, S., Dabney, W., Valko, M., and Abel, D. Revisiting Peng's Q(λ) for modern reinforcement learning. *arXiv:2103.00107*, 2021.

Lin, L.-J. Self-improving reactive agents based on reinforcement learning, planning and reaching. *Machine Learning*, 8: 293–321, 1992.

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

Munos, R., Stepleton, T., Harutyunyan, A., and Bellemare, M. G. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, 2016.

Patterson, A., White, A., and White, M. A generalized projected Bellman error for off-policy value estimation in reinforcement learning. *Journal of Machine Learning Research*, 2022.

Peng, J. and Williams, R. J. Incremental multi-step Q-Learning. *Machine Learning*, 22:226–232, 1996.

Precup, D., Sutton, R. S., and Singh, S. Eligibility traces for off-policy policy evaluation. In *International Conference on Machine Learning*, pp. 759–766, 2000.

Rowland, M., Dabney, W., and Munos, R. Adaptive trade-offs in off-policy learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 34–44, 2020.

Rummery, G. A. and Niranjan, M. On-line Q-Learning using connectionist systems. Technical report, Cambridge University, 1994.

Singh, S. P. and Sutton, R. S. Reinforcement learning with replacing eligibility traces. *Machine Learning*, 22:123–158, 1996.

Sutton, R. S. *Temporal Credit Assignment in Reinforcement Learning*. PhD thesis, University of Massachusetts Amherst, 1984.

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

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. MIT Press, 1st edition, 1998.

Sutton, R. S., Maei, H. R., Precup, D., Bhatnagar, S., Silver, D., Szepesvári, C., and Wiewiora, E. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *International Conference on Machine Learning*, pp. 993–1000, 2009.

Touati, A., Bacon, P.-L., Precup, D., and Vincent, P. Convergent Tree Backup and Retrace with function approximation. In *International Conference on Machine Learning*, pp. 4955–4964, 2018.

Uchibe, E. and Doya, K. Competitive-cooperative-concurrent reinforcement learning with importance sampling. In *International Conference on Simulation of Adaptive Behavior: From Animals and Animats*, pp. 287–296, 2004.

Wang, Z., Bapst, V., Heess, N., Mnih, V., Munos, R., Kavukcuoglu, K., and de Freitas, N. Sample efficient actor-critic with experience replay. In *International Conference on Learning Representations*, 2017.

Watkins, C. J. C. H. *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, 1989.

Wawrzyński, P. Real-time reinforcement learning by sequential actor-critics and experience replay. *Neural Networks*, 22 (10):1484–1497, 2009.

Wawrzyński, P. and Pacut, A. Truncated importance sampling for reinforcement learning with experience replay. *International Multiconference on Computer Science and Information Technology*, pp. 305–315, 2007.

Yu, H., Mahmood, A. R., and Sutton, R. S. On generalized Bellman equations and temporal-difference learning. *Journal of Machine Learning Research*, 19(1):1864–1912, 2018.

A. $\mathcal{M}$ Operator Details

In Section 5, we defined a linear operator B_t, where

$$(B_t X)(s,a) = \mathbb{E}_\mu\big[\beta_t X(S_t, A_t) \;\big|\; (S_0, A_0) = (s,a)\big], \tag{6}$$

such that the expected-value version of our $\mathcal{M}$ operator,

$$(\mathcal{M}Q)(s,a) = Q(s,a) + \mathbb{E}_\mu\left[\sum_{t=0}^{\infty} \gamma^t \beta_t \delta_t^\pi \;\middle|\; (S_0, A_0) = (s,a)\right] \tag{4}$$

$$= Q(s,a) + \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_\mu\big[\beta_t \delta_t^\pi \;\big|\; (S_0, A_0) = (s,a)\big], \tag{5}$$

is element-wise equivalent to the vector version,

$$\mathcal{M}Q = Q + \sum_{t=0}^{\infty} \gamma^t B_t (T_\pi Q - Q). \tag{7}$$

We claimed that each element of B_t must have the form

$$B_t((s,a),(s',a')) = \Pr_\mu((S_t, A_t) = (s',a') \mid (S_0, A_0) = (s,a)) \times \mathbb{E}_\mu\big[\beta_t \;\big|\; (S_0, A_0) = (s,a), (S_t, A_t) = (s',a')\big], \tag{8}$$

with (s,a) as the row index and (s',a') as the column index. This is because multiplying this matrix B_t with a vector X results in the same operation as the weighted expected value in Eq. (5):

$$\begin{aligned}
\sum_{s',a'} B_t((s,a),(s',a'))X(s',a') &= \mathbb{E}_\mu\left[\mathbb{E}_\mu\big[\beta_t \;\big|\; (S_0, A_0) = (s,a), (S_t, A_t)\big] \cdot X(S_t, A_t) \;\middle|\; (S_0, A_0) = (s,a)\right] \\
&= \mathbb{E}_\mu\left[\mathbb{E}_\mu\big[\beta_t X(S_t, A_t) \;\big|\; (S_0, A_0) = (s,a), (S_t, A_t)\big] \;\middle|\; (S_0, A_0) = (s,a)\right] \\
&= \mathbb{E}_\mu\big[\beta_t X(S_t, A_t) \;\big|\; (S_0, A_0) = (s,a)\big].
\end{aligned} \tag{14}$$

So, when X is the expected TD error $T_\pi Q - Q$, Eq. (7) becomes Eq. (5) exactly.

$\mathcal{M}$ is a contraction mapping whenever $\beta_t \leq \beta_{t-1}\rho_t$ for all t (Condition 5.1), which Theorem 5.2 establishes. As we discussed in Section 5.3, violating this condition can sometimes cause $\mathcal{M}$ to no longer contract, even with on-policy updates. We can see one plausible reason for this by refactoring the definition of $\mathcal{M}$. Let $q_t \coloneqq Q(S_t, A_t)$ and $v_t \coloneqq \sum_{a' \in \mathcal{A}} \pi(a'|S_t) Q(S_t, a')$, so $\delta_t^\pi = R_t + \gamma v_{t+1} - q_t$. Further, assume the following expectations are conditioned on $(S_0, A_0) = (s,a)$. Eq. (4) is equivalent to

$$\begin{aligned}
(\mathcal{M}Q)(s,a) &= q_0 + \mathbb{E}_\mu\left[\sum_{t=0}^{\infty} \gamma^t \beta_t (R_t + \gamma v_{t+1} - q_t)\right] \\
&= q_0 + \mathbb{E}_\mu\left[\sum_{t=0}^{\infty} \gamma^t \beta_t R_t + \sum_{t=1}^{\infty} \gamma^t \beta_{t-1} v_t - \sum_{t=0}^{\infty} \gamma^t \beta_t q_t\right] \\
&= \mathbb{E}_\mu\left[\sum_{t=0}^{\infty} \gamma^t \beta_t R_t + \sum_{t=1}^{\infty} \gamma^t \beta_{t-1} v_t - \sum_{t=1}^{\infty} \gamma^t \beta_t q_t\right] \\
&= \mathbb{E}_\mu\left[\sum_{t=0}^{\infty} \gamma^t \beta_t R_t\right] + \mathbb{E}_\mu\left[\sum_{t=1}^{\infty} \gamma^t (\beta_{t-1} v_t - \beta_t q_t)\right] \\
&= \mathbb{E}_\mu\left[\sum_{t=0}^{\infty} \gamma^t \beta_t R_t\right] + \mathbb{E}_\mu\left[\sum_{t=1}^{\infty} \gamma^t (\beta_{t-1}\rho_t - \beta_t) q_t\right],
\end{aligned} \tag{15}$$

and we discussed in Section 5.3 that these two terms represent a biased return estimate and an infinite sum of weighted value-function bootstraps, respectively. In particular, this can be problematic if $\beta_t > \beta_{t-1}\rho_t$ because the corresponding bootstrap's weight becomes negative, causing it to get subtracted from the return estimate.

B. Additional Proofs

B.1. Proof of Lemma B.1

Lemma B.1. *Q^π is a fixed point of $\mathcal{M}$; the difference between $\mathcal{M}Q$ and Q^π is given by*

$$\mathcal{M}Q - Q^\pi = Z(Q - Q^\pi), \tag{16}$$

where $Z \coloneqq \sum_{t=1}^{\infty} \gamma^t (B_{t-1}P_\pi - B_t)$.

Proof. It is evident from Eq. (7) that Q^π is a fixed point of $\mathcal{M}$ because $T_\pi Q^\pi - Q^\pi = 0$, and so $\mathcal{M}Q^\pi = Q^\pi$. Therefore,

$$\begin{aligned}
\mathcal{M}Q - Q^\pi &= \mathcal{M}Q - \mathcal{M}Q^\pi \\
&= Q + \sum_{t=0}^{\infty} \gamma^t B_t (T_\pi Q - Q) - Q^\pi - \sum_{t=0}^{\infty} \gamma^t B_t (T_\pi Q^\pi - Q^\pi) \\
&= Q - Q^\pi + \sum_{t=0}^{\infty} \gamma^t B_t (T_\pi Q - T_\pi Q^\pi) - \sum_{t=0}^{\infty} \gamma^t B_t (Q - Q^\pi) \\
&= \sum_{t=0}^{\infty} \gamma^t B_t (T_\pi Q - T_\pi Q^\pi) - \sum_{t=1}^{\infty} \gamma^t B_t (Q - Q^\pi) \\
&= \sum_{t=0}^{\infty} \gamma^{t+1} B_t P_\pi (Q - Q^\pi) - \sum_{t=1}^{\infty} \gamma^t B_t (Q - Q^\pi) \\
&= \left(\sum_{t=0}^{\infty} \gamma^{t+1} B_t P_\pi - \sum_{t=1}^{\infty} \gamma^t B_t \right) (Q - Q^\pi) \\
&= \left(\sum_{t=1}^{\infty} \gamma^t (B_{t-1} P_\pi - B_t) \right) (Q - Q^\pi) \\
&= Z(Q - Q^\pi),
\end{aligned}$$

which is the desired result. □

B.2. Proof of Lemma B.2

Lemma B.2. *If Condition 5.1 holds, then Z has nonnegative elements and its row sums obey $Z\mathbf{1} \leq \gamma$.*

Proof. Define the linear operator $D_t \coloneqq B_{t-1}P_\pi - B_t$ and notice that $Z = \sum_{t=1}^{\infty} \gamma^t D_t$. We will show that D_t comprises only nonnegative elements, and therefore so does Z. For any $X \in \mathbb{R}^n$, observe that

$$\begin{aligned}
(D_t X)(s,a) &= \mathbb{E}_\mu \left[\beta_{t-1} \sum_{S_t \in \mathcal{S}} \sum_{A_t \in \mathcal{A}} P(S_t|\mathcal{F}_{t-1}) \pi(A_t|S_t) X(S_t, A_t) \;\middle|\; (S_0, A_0) = (s,a) \right] \\
&\quad - \mathbb{E}_\mu \big[\beta_t X(S_t, A_t) \;\big|\; (S_0, A_0) = (s,a) \big] \\
&= \mathbb{E}_\mu \left[\beta_{t-1} \sum_{S_t \in \mathcal{S}} \sum_{A_t \in \mathcal{A}} P(S_t|\mathcal{F}_{t-1}) \pi(A_t|S_t) X(S_t, A_t) \;\middle|\; (S_0, A_0) = (s,a) \right] \\
&\quad - \mathbb{E}_\mu \left[\sum_{S_t \in \mathcal{S}} \sum_{A_t \in \mathcal{A}} P(S_t|\mathcal{F}_{t-1}) \mu(A_t|S_t) \beta_t X(S_t, A_t) \;\middle|\; (S_0, A_0) = (s,a) \right] \\
&= \mathbb{E}_\mu \left[\sum_{S_t \in \mathcal{S}} P(S_t|\mathcal{F}_{t-1}) \sum_{A_t \in \mathcal{A}} \big(\pi(A_t|S_t)\beta_{t-1} - \mu(A_t|S_t)\beta_t \big) X(S_t, A_t) \;\middle|\; (S_0, A_0) = (s,a) \right].
\end{aligned} \tag{17}$$

Since we assumed that $\beta_t \leq \beta_{t-1}\rho_t$ in Condition 5.1, we have $\pi(A_t|S_t)\beta_{t-1} - \mu(A_t|S_t)\beta_t \geq 0$, which implies that $D_t \geq 0$. Furthermore, this holds for all $t \geq 1$, so $Z \geq 0$ follows immediately.

To complete the proof, we show that the row sums of Z are bounded by γ. Recall that $P_\pi \mathbf{1} = \mathbf{1}$. Hence,

$$\begin{aligned}
Z\mathbf{1} &= \sum_{t=1}^{\infty} \gamma^t (B_{t-1} P_\pi - B_t)\mathbf{1} \\
&= \sum_{t=1}^{\infty} \gamma^t (B_{t-1}\mathbf{1} - B_t\mathbf{1}) \\
&= \sum_{t=0}^{\infty} \gamma^{t+1} B_t \mathbf{1} - \sum_{t=1}^{\infty} \gamma^t B_t \mathbf{1} \\
&= \gamma\mathbf{1} + \sum_{t=1}^{\infty} \gamma^{t+1} B_t \mathbf{1} - \sum_{t=1}^{\infty} \gamma^t B_t \mathbf{1} \\
&= \gamma\mathbf{1} - (1-\gamma) \sum_{t=1}^{\infty} \gamma^t B_t \mathbf{1} \\
&\leq \gamma\mathbf{1}, \qquad (18)
\end{aligned}$$

because $B_t \geq 0, \forall\, t \geq 1$. □

B.3. Proof of Theorem 5.3

Theorem 5.3. *Consider a sequence of target policies $(\pi_i)_{i\geq 0}$ and a sequence of arbitrary behavior policies $(\mu_i)_{i\geq 0}$. Let Q_0 be an arbitrary vector in $\mathbb{R}^n$ and define the sequence $Q_{i+1} \coloneqq \mathcal{M}_i Q_i$, where $\mathcal{M}_i$ is the operator defined by Eq. (10). Assume that $(\pi_i)_{i\geq 0}$ is greedy in the limit, and let $\epsilon_i \geq 0$ be the smallest constant such that $T_{\pi_i} Q_i \geq TQ_i - \epsilon_i \|Q_i\| \mathbf{1}$. If Condition 5.1 holds for all i, then*

$$\|\mathcal{M}_i Q_i - Q^*\| \leq \gamma \|Q_i - Q^*\| + \frac{\epsilon_i}{1-\gamma} \|Q_i\|, \qquad (11)$$

and, consequently, $\lim_{i\to\infty} Q_i = Q^*$.

Proof. We first derive the following upper bound:

$$T_{\pi_i} Q_i - TQ^* = \gamma P_{\pi_i} Q_i - \gamma \max_\pi P_\pi Q^* \leq \gamma P_{\pi_i}(Q_i - Q^*). \qquad (19)$$

From Eq. (10) and because C_i has nonnegative entries, we can deduce that

$$\begin{aligned}
\mathcal{M}_i Q_i - Q^* &= (I - C_i)(Q_i - Q^*) + C_i(T_{\pi_i} Q_i - Q^*) \qquad (20) \\
&= (I - C_i)(Q_i - Q^*) + C_i(T_{\pi_i} Q_i - TQ^*) \\
&\leq (I - C_i)(Q_i - Q^*) + \gamma C_i P_{\pi_i}(Q_i - Q^*) \\
&= Z_i(Q_i - Q^*), \qquad (21)
\end{aligned}$$

where $Z_i \coloneqq I - C_i(I - \gamma P_{\pi_i})$. Notice that Z_i is analogous to the matrix Z in Eq. (9) because, for policies π_i and μ_i,

$$\begin{aligned}
I - C_i(I - \gamma P_{\pi_i}) &= I + \sum_{t=0}^{\infty} \gamma^t B_t (\gamma P_{\pi_i} - I) \\
&= I + \sum_{t=0}^{\infty} \gamma^{t+1} B_t P_{\pi_i} - \sum_{t=0}^{\infty} \gamma^t B_t \\
&= \sum_{t=1}^{\infty} \gamma^t B_{t-1} P_{\pi_i} - \sum_{t=1}^{\infty} \gamma^t B_t \\
&= \sum_{t=1}^{\infty} \gamma^t (B_{t-1} P_{\pi_i} - B_t). \qquad (22)
\end{aligned}$$

Next, we derive the following lower bound:

$$TQ_i - TQ^* \geq T_{\pi^*}Q_i - TQ^* = \gamma P_{\pi^*}(Q_i - Q^*). \tag{23}$$

Additionally, for each policy π_i, there exists some $\epsilon_i \geq 0$ such that $T_{\pi_i}Q_i \geq TQ_i - \epsilon_i\|Q_i\|\mathbf{1}$ (recall that we defined ϵ_i to be as small as possible). Starting again from Eq. (20), and noting that the elements of C_i are nonnegative, we obtain

$$\begin{aligned}
\mathcal{M}_iQ_i - Q^* &\geq (I - C_i)(Q_i - Q^*) + C_i(TQ_i - Q^*) - \epsilon_i\|Q_i\|C_i\mathbf{1} \\
&= (I - C_i)(Q_i - Q^*) + C_i(TQ_i - TQ^*) - \epsilon_i\|Q_i\|C_i\mathbf{1} \\
&\geq (I - C_i)(Q_i - Q^*) + \gamma C_iP_{\pi^*}(Q_i - Q^*) - \epsilon_i\|Q_i\|C_i\mathbf{1} \\
&= Z_i^*(Q_i - Q^*) - \epsilon_i\|Q_i\|C_i\mathbf{1},
\end{aligned} \tag{24}$$

where we have defined $Z_i^* := I - C_i(I - \gamma P_{\pi^*})$. By Lemma B.2, since we assumed Condition 5.1 holds, both Z_i and Z_i^* have nonnegative elements and their row sums are bounded by γ. Therefore, when $\mathcal{M}_iQ_i - Q^* \geq 0$, Eq. (21) implies

$$\|\mathcal{M}_iQ_i - Q^*\| \leq \gamma\|Q_i - Q^*\|, \tag{25}$$

because element-wise inequality for nonnegative matrices implies the inequality holds also for their norms. When $\mathcal{M}_iQ_i - Q^* \leq 0$, we must use Eq. (24) and multiply both sides by -1 to get nonnegative matrices, giving

$$\begin{aligned}
\|\mathcal{M}_iQ_i - Q^*\| &\leq \gamma\|Q_i - Q^*\| + \epsilon_i\|Q_i\|\|C_i\| \\
&\leq \gamma\|Q_i - Q^*\| + \frac{\epsilon_i}{1-\gamma}\|Q_i\|,
\end{aligned} \tag{26}$$

because $\|C_i\| \leq \sum_{t=0}^{\infty}\gamma^t\|P_{\pi_i}\|^t = (1-\gamma)^{-1}$. Since Eq. (26) is looser than Eq. (25), its bound holds in the worst case. It remains to show that this bound implies convergence to Q^*. Observe that

$$\begin{aligned}
\gamma\|Q_i - Q^*\| + \frac{\epsilon_i}{1-\gamma}\|Q_i\| &\leq \gamma\|Q_i - Q^*\| + \frac{\epsilon_i}{1-\gamma}(\|Q_i - Q^*\| + \|Q^*\|) \\
&= \left(\gamma + \frac{\epsilon_i}{1-\gamma}\right)\|Q_i - Q^*\| + \frac{\epsilon_i}{1-\gamma}\|Q^*\|.
\end{aligned} \tag{27}$$

Our assumption of greedy-in-the-limit policies tells us that $\epsilon_i \to 0$ as $i \to \infty$; thus, there must exist some iteration i^* such that $\epsilon_i \leq \frac{1}{2}(1-\gamma)^2, \forall\, i \geq i^*$. Therefore, for $i \geq i^*$,

$$\|\mathcal{M}_iQ_i - Q^*\| \leq \frac{1+\gamma}{2}\|Q_i - Q^*\| + \frac{\epsilon_i}{1-\gamma}\|Q^*\|. \tag{28}$$

If $\gamma < 1$, then $\frac{1}{2}(1+\gamma) < 1$, and since $\|Q^*\|$ is finite, we conclude that $\|Q_i - Q^*\| \to 0$ as $i \to \infty$. □

C. Examples of Divergence

C.1. Counterexample 5.7: Off-Policy Truncated IS

Our definitions of π and μ give us

$$P_\pi = \begin{bmatrix} p & 1-p \\ p & 1-p \end{bmatrix}, \qquad P_\mu = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}. \tag{29}$$

Recall that we assumed $\lambda = 1$. We define the following constant, using the definition of β_t for Truncated IS:

$$\begin{aligned}
\beta_t^{(1)} &:= \mathbb{E}\big[\beta_t \;\big|\; (S_t, A_t) = (s, a_1)\big] \\
&= \sum_{\mathcal{F}_t}\Pr_\mu(\mathcal{F}_t \mid (S_t, A_t) = (s, a_1)) \cdot \min\left(1,\ \frac{\Pr_\pi(\mathcal{F}_t)}{\Pr_\mu(\mathcal{F}_t)}\right) \\
&= \sum_{\mathcal{F}_{t-1}}\Pr_\mu(\mathcal{F}_{t-1})\min\left(1,\ \frac{\Pr_\pi(\mathcal{F}_{t-1})\cdot p}{\Pr_\mu(\mathcal{F}_{t-1})\cdot\frac{1}{2}}\right)
\end{aligned} \tag{30}$$

$$= \sum_{\mathcal{F}_{t-1}} \min\left(\Pr_\mu(\mathcal{F}_{t-1}),\ 2p \cdot \Pr_\pi(\mathcal{F}_{t-1})\right)$$

$$= \sum_{\mathcal{F}_{t-1}} \min\left(\frac{1}{2^{t-1}},\ 2p \cdot \Pr_\pi(\mathcal{F}_{t-1})\right). \tag{31}$$

Eq. (30) is justified because the conditional probability of a trajectory ending in action a_1 is just the probability of $\mathcal{F}_{t-1}$ under μ, due to the 1-state (memoryless) MDP. We can simplify $\beta_t^{(1)}$ further by using the binomial theorem to calculate $\Pr_\pi(\mathcal{F}_{t-1}) = p^k(1-p)^{t-1-k}$, where $k \in [0, t-1]$ is the number of times a_1 is taken in $\mathcal{F}_{t-1}$. There are $\binom{t-1}{k}$ trajectories with this same probability. Therefore,

$$\beta_t^{(1)} = \sum_{\mathcal{F}_{t-1}} \min\left(\frac{1}{2^{t-1}},\ 2p \cdot \Pr_\pi(\mathcal{F}_{t-1})\right) = \sum_{k=0}^{t-1} \binom{t-1}{k} \min\left(\frac{1}{2^{t-1}},\ 2p \cdot p^k(1-p)^{t-1-k}\right). \tag{32}$$

Likewise, we can compute $\beta_t^{(2)}$ by swapping p and $1-p$ above. Let $\odot$ denote element-wise multiplication. Using the fact that $P_\mu^t = P_\mu, \forall\, t \geq 1$, it follows that

$$B_t = P_\mu^t \odot \begin{bmatrix} \beta_t^{(1)} & \beta_t^{(2)} \\ \beta_t^{(1)} & \beta_t^{(2)} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \beta_t^{(1)} & \beta_t^{(2)} \\ \beta_t^{(1)} & \beta_t^{(2)} \end{bmatrix}. \tag{33}$$

Using a computer program to calculate Z, assuming that $p = 0.6$ and $\gamma = 0.94$, we obtain

$$Z = \sum_{t=1}^{\infty} \gamma^t (B_{t-1}P_\pi - B_t) \approx \begin{bmatrix} 0.704 & -0.436 \\ 0.704 & -0.436 \end{bmatrix}. \tag{34}$$

Therefore, $\|Z\| \approx 1.14$, which is not a contraction, and the norm continues to increase for $p > 0.6$ or $\gamma > 0.94$.

C.2. Counterexample 5.8: On-Policy Binary Traces

The policy π is uniform random, so we have

$$P_\pi = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}. \tag{35}$$

Let $\odot$ denote element-wise multiplication. Because $\beta_t = 1$ only when the trajectory $\mathcal{F}_t$ terminates in (s, a_1) and $\beta_t = 0$ otherwise, and since $P_\pi^t = P_\pi, \forall\, t \geq 1$, we also have

$$B_t = P_\pi^t \odot \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}. \tag{36}$$

Using a computer program to calculate Z, assuming that $\gamma = \frac{2}{3}$, we obtain

$$Z = \sum_{t=1}^{\infty} \gamma^t (B_{t-1}P_\pi - B_t) = \frac{1}{3} \begin{bmatrix} -1 & 2 \\ -1 & 2 \end{bmatrix}. \tag{37}$$

Therefore, $\|Z\| = 1$, which is not a contraction, and the norm continues to increase for $\gamma > \frac{2}{3}$.

D. Implementation of Trajectory-Aware Eligibility Traces

The implementation of trajectory-aware methods is closely related to that of backward-view TD(λ) in the tabular setting (see, e.g., Sutton & Barto, 1998, Chapter 7.3). On each timestep, an environment interaction is conducted according to the behavior policy μ. Then, the eligibilities for previously visited state-action pairs are modified, the eligibility for the current state-action pair is incremented, and the current TD error is applied to all state-action pairs in proportion to their eligibilities. The only difference in the trajectory-aware case is that the eligibilities are not modified by simply multiplying a constant decay factor $\gamma\lambda$.

Arbitrary, trajectory-dependent traces $\beta(\mathcal{F}_t)$, as studied in our theoretical results, can be complicated to implement. This stems from the fact that the timestep t in the $\mathcal{M}$ operator is defined *relative* to when the updated state-action pair was taken. In other words, each state-action pair (S_k, A_k) "disagrees" on the start of the current trajectory, generating its update from the unique sub-trajectory $(S_k, A_k), \ldots, (S_t, A_t)$. Implementing coefficients of this form would be possible using the general update

$$Q(S_k, A_k) \leftarrow Q(S_k, A_k) + \alpha\gamma^{t-k}\beta((S_k, A_k), \ldots, (S_t, A_t))\delta_t^{\pi}, \tag{38}$$

where $\alpha \in (0, 1]$ is the stepsize, but this would require repeatedly slicing the list of visited state-action pairs $(S_0, A_0), \ldots, (S_t, A_t)$. While this is certainly feasible, it does not easily accommodate vectorization or parallelization.

Fortunately, this level of generality is rarely needed in practice, and specific optimizations can be made depending on the functional form of β. For example, Truncated IS defines β to be a pure function of the IS estimate Π_t, which is useful because per-decision eligibility traces can be used to efficiently generate the IS estimates for every state-action pair visited during the episode. We demonstrate how this can be done in pseudocode (see Algorithm 1).

Recursive methods like Recursive Retrace and RBIS, where β_t explicitly depends on β_{t-1}, require only two minor changes compared to Algorithm 1 for their implementations. These changes, which we highlight in blue for RBIS in Algorithm 2, correspond to the fact that the dynamic array Y is now used to store the previous trace β_{t-1} rather than the previous IS estimate Π_{t-1} at each timestep. The computational requirements for the methods remain nearly identical. The implementation for Recursive Retrace easily follows by changing line 10 of Algorithm 2 to

$$Y(k) \leftarrow \lambda \min(1,\ Y(k) \cdot \rho_t). \tag{39}$$

E. Additional Experiment Details and Results

We conducted a grid search to find the best stepsize α for every λ-value for the four off-policy methods we evaluated in the Bifurcated Gridworld (Section 6). Using a training set of 1,000 trials, we searched over $\lambda \in \{0, 0.1, \ldots, 1\}$ and $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, for a total of 55 hyperparameter combinations. At the start of each trial, the initial value function Q was sampled from a zero-mean Gaussian distribution with standard deviation $\sigma = 0.01$. We trained each agent for 3,000 timesteps, allowing extra time to complete the final episode. We then generated learning curves by plotting the 100-episode moving average of these returns as a function of the number of timesteps and calculated their AUCs. In Table 1, we report the stepsize α that led to the highest average AUC for each λ-value. Then, using a separate test set of 1,000 trials to avoid bias in the search results, these α-values were used to generate the learning curves in Figure 3. The AUCs for these learning curves were finally used in the creation of the λ-sweep plot (Figure 2).

In Figure 4, we repeated this procedure for three additional, more complex gridworld topologies. Like the Bifurcated Gridworld, these environments feature one or more bifurcations that make fast credit assignment imperative, as well as additional challenges such as multiple goal cells. As before, the agent starts in S and receives $+1$ reward for taking any action in a goal, terminating the episode. The results are qualitatively similar to Figure 2; RBIS outperforms the other three methods by a significant margin over the left-hand portion of the λ-spectrum, and performs similarly to Retrace as $\lambda \to 1$.

Table 1. The best stepsizes found by our grid search in the Bifurcated Gridworld.

λ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Retrace	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.7	0.7	0.5
Truncated IS	0.9	0.9	0.9	0.9	0.9	0.9	0.7	0.5	0.5	0.5	0.3
Recursive Retrace	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.7	0.5	0.5
RBIS	0.9	0.9	0.9	0.9	0.9	0.7	0.7	0.7	0.7	0.7	0.5

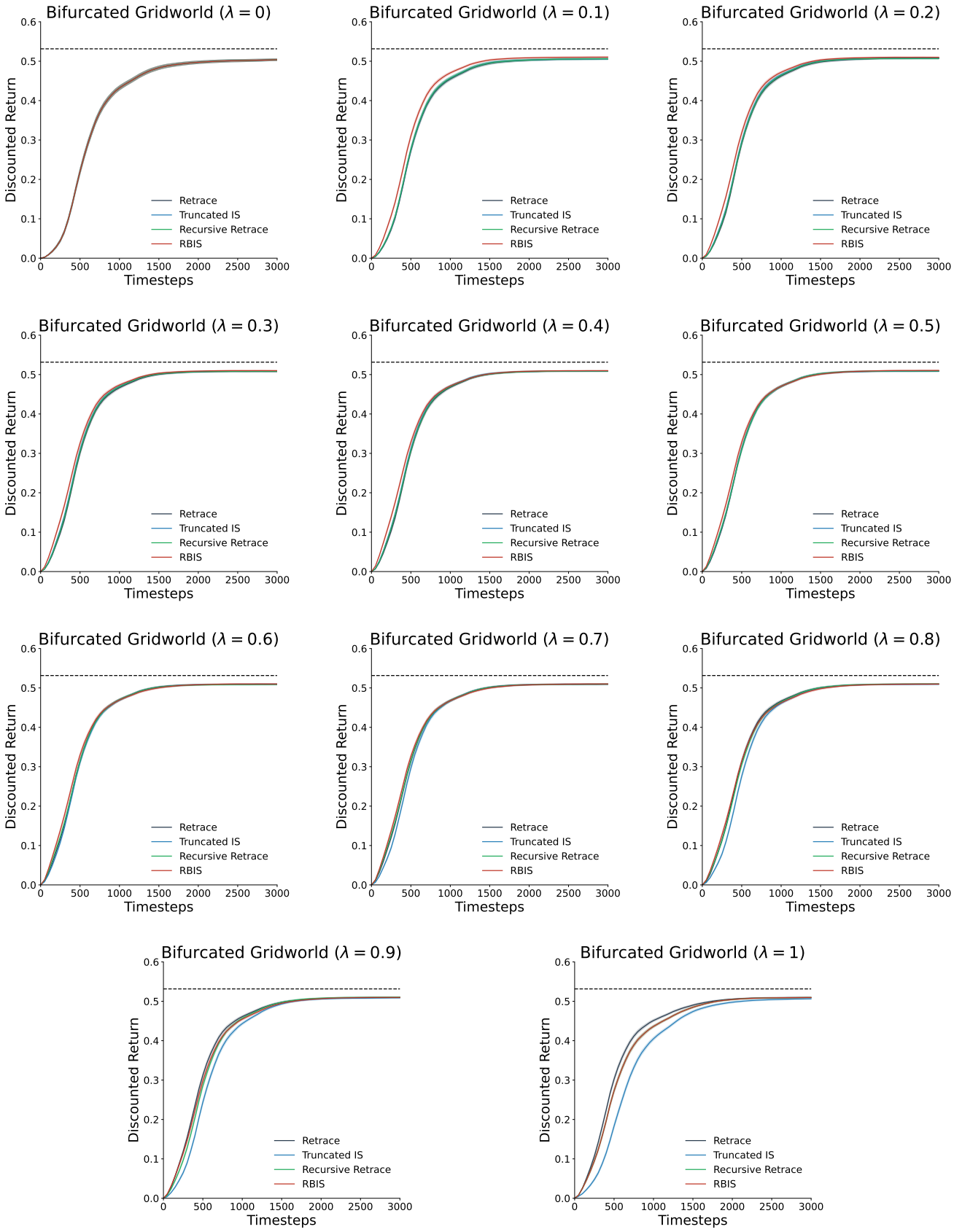

Figure 3. Learning curves for the λ-values we tested in the Bifurcated Gridworld environment. The dashed black line indicates the optimal discounted return for this problem.

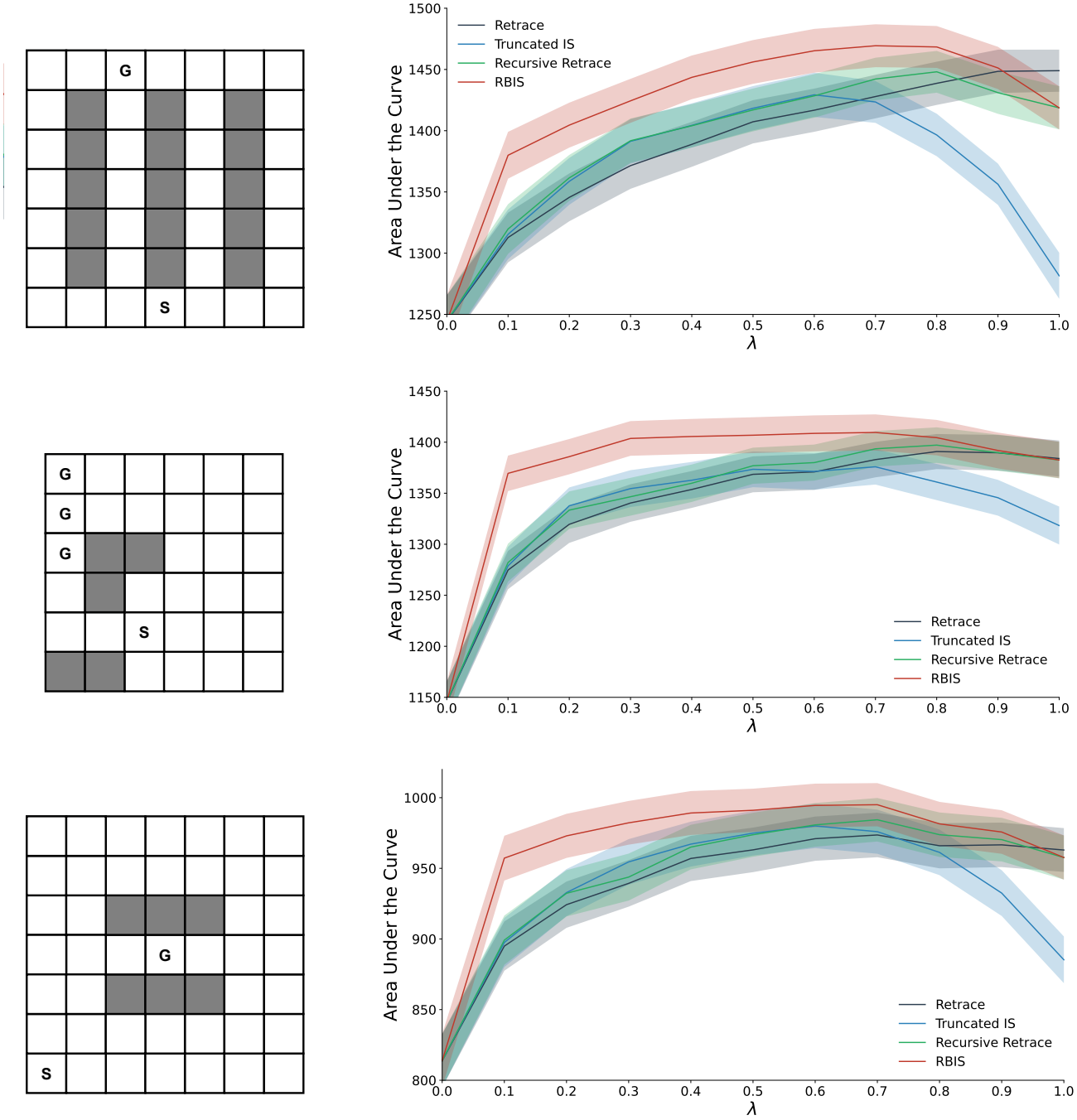

Figure 4. λ-sweeps conducted on three additional gridworld topologies. The experiment procedure was identical to that used in the creation of Figure 2.

Algorithm 1 Truncated Importance Sampling

Input: value function Q, stepsize $\alpha \in (0, 1]$
for each episode **do**
  Reset environment and observe state S_0
  Reset dynamic array Y
  repeat $\{$for $t = 0, 1, 2, \dots\}$
    Take action $A_t \sim \mu(\cdot|S_t)$, receive reward R_t, and observe next state S_{t+1}
    $\rho_t = \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)}$
    $\delta_t = \begin{cases} R_t - Q(S_t, A_t) & \text{if } S_{t+1} \text{ is terminal} \\ R_t - Q(S_t, A_t) + \gamma \sum_{a' \in \mathcal{A}} \pi(a'|S_{t+1}) Q(S_{t+1}, a') & \text{else} \end{cases}$
    for $k = 0, \dots, t-1$ **do**
      $Y(k) \leftarrow Y(k) \cdot \rho_t$
    end for
    $Y(t) \leftarrow 1$
    for $k = 0, \dots, t$ **do**
      $z \leftarrow (\gamma\lambda)^{t-k} \min(1,\ Y(k))$
      $Q(S_k, A_k) \leftarrow Q(S_k, A_k) + \alpha z \delta_t$
    end for
  until S_{t+1} is terminal
end for

Algorithm 2 Recency-Bounded Importance Sampling (RBIS)

Input: value function Q, stepsize $\alpha \in (0, 1]$
for each episode **do**
  Reset environment and observe state S_0
  Reset dynamic array Y
  repeat $\{$for $t = 0, 1, 2, \dots\}$
    Take action $A_t \sim \mu(\cdot|S_t)$, receive reward R_t, and observe next state S_{t+1}
    $\rho_t = \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)}$
    $\delta_t = \begin{cases} R_t - Q(S_t, A_t) & \text{if } S_{t+1} \text{ is terminal} \\ R_t - Q(S_t, A_t) + \gamma \sum_{a' \in \mathcal{A}} \pi(a'|S_{t+1}) Q(S_{t+1}, a') & \text{else} \end{cases}$
    for $k = 0, \dots, t-1$ **do**
      $Y(k) \leftarrow \min(\lambda^{t-k},\ Y(k) \cdot \rho_t)$
    end for
    $Y(t) \leftarrow 1$
    for $k = 0, \dots, t$ **do**
      $z \leftarrow \gamma^{t-k} Y(k)$
      $Q(S_k, A_k) \leftarrow Q(S_k, A_k) + \alpha z \delta_t$
    end for
  until S_{t+1} is terminal
end for