Importance Resampling for Off-policy Prediction

Matthew Schlegel
University of Alberta
mkschleg@ualberta.ca

Wesley Chung
University of Alberta
wchung@ualberta.ca

Daniel Graves
Huawei
daniel.graves@huawei.com

Jian Qian
University of Alberta
jq1@ulberta.ca

Martha White
University of Alberta
whitem@ulberta.ca

Abstract

Importance sampling (IS) is a common reweighting strategy for off-policy prediction in reinforcement learning. While it is consistent and unbiased, it can result in high variance updates to the weights for the value function. In this work, we explore a resampling strategy as an alternative to reweighting. We propose Importance Resampling (IR) for off-policy prediction, which resamples experience from a replay buffer and applies standard on-policy updates. The approach avoids using importance sampling ratios in the update, instead correcting the distribution before the update. We characterize the bias and consistency of IR, particularly compared to Weighted IS (WIS). We demonstrate in several microworlds that IR has improved sample efficiency and lower variance updates, as compared to IS and several variance-reduced IS strategies, including variants of WIS and V-trace which clips IS ratios. We also provide a demonstration showing IR improves over IS for learning a value function from images in a racing car simulator.

1 Introduction

An emerging direction for reinforcement learning systems is to learn many predictions, formalized as value function predictions contingent on many different policies. The idea is that such predictions can provide a powerful abstract model of the world. Some examples of systems that learn many value functions are the Horde architecture composed of General Value Functions (GVFs) [Sutton et al., 2011, Modayil et al., 2014], systems that use options [Sutton et al., 1999, Schaul et al., 2015a], predictive representation approaches [Sutton et al., 2005, Schaul and Ring, 2013, Silver et al., 2017] and systems with auxiliary tasks [Jaderberg et al., 2017]. Off-policy learning is critical for learning many value functions with different policies, because it enables data to be generated from one behavior policy to update the values for each target policy in parallel.

The typical strategy for off-policy learning is to reweight updates using importance sampling (IS). For a given state s, with action a selected according to behavior μ, the IS ratio is the ratio between the probability of the action under the target policy π and the behavior: $\frac{\pi(a|s)}{\mu(a|s)}$. The update is multiplied by this ratio, adjusting the action probabilities so that the expectation of the update is as if the actions were sampled according to the target policy π. Though the IS estimator is unbiased and consistent [Kahn and Marshall, 1953, Rubinstein and Kroese, 2016], it can suffer from high or even infinite variance due to large magnitude IS ratios, in theory [Andradottir et al., 1995] and in practice [Precup et al., 2001, Mahmood et al., 2014, 2017].

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

There have been some attempts to modify off-policy prediction algorithms to mitigate this variance.[1] Weighted IS (WIS) algorithms have been introduced [Precup et al., 2001, Mahmood et al., 2014, Mahmood and Sutton, 2015], which normalize each update by the sample average of the ratios. These algorithms improve learning over standard IS strategies, but are not straightforward to extend to nonlinear function approximation. In the offline setting, a reweighting scheme, called importance sampling with unequal support [Thomas and Brunskill, 2017], was introduced to account for samples where the ratio is zero, in some cases significantly reducing variance. Another strategy is to rescale or truncate the IS ratios, as used by V-trace [Espeholt et al., 2018] for learning value functions and Tree-Backup [Precup et al., 2000], Retrace [Munos et al., 2016] and ABQ [Mahmood et al., 2017] for learning action-values. Truncation of IS-ratios in V-trace can incur significant bias, and this additional truncation parameter needs to be tuned.

An alternative to reweighting updates is to instead correct the distribution before updating the estimator using weighted bootstrap sampling: resampling a new set of data from the previously generated samples [Smith et al., 1992, Arulampalam et al., 2002]. Consider a setting where a buffer of data is stored, generated by a behavior policy. Samples for policy π can be obtained by resampling from this buffer, proportionally to $\frac{\pi(a|s)}{\mu(a|s)}$ for state-action pairs (s, a) in the buffer. In the sampling literature, this strategy has been proposed under the name Sampling Importance Resampling (SIR) [Rubin, 1988, Smith et al., 1992, Gordon et al., 1993], and has been particularly successful for Sequential Monte Carlo sampling [Gordon et al., 1993, Skare et al., 2003]. Such resampling strategies have also been popular in classification, with over-sampling or under-sampling typically being preferred to weighted (cost-sensitive) updates [Lopez et al., 2013].

A resampling strategy has several potential benefits for off-policy prediction.[2] Resampling could even have larger benefits for learning approaches, as compared to averaging or numerical integration problems, because updates accumulate in the weight vector and change the optimization trajectory of the weights. For example, very large importance sampling ratios could destabilize the weights. This problem does not occur for resampling, as instead the same transition will be resampled multiple times, spreading out a large magnitude update across multiple updates. On the other extreme, with small ratios, IS will waste updates on transitions with very small IS ratios. By correcting the distribution before updating, standard on-policy updates can be applied. The magnitude of the updates vary less—because updates are not multiplied by very small or very large importance sampling ratios—potentially reducing variance of stochastic updates and simplifying learning rate selection. We hypothesize that resampling (a) learns in a fewer number of updates to the weights, because it focuses computation on samples that are likely under the target policy and (b) is less sensitive to learning parameters and target and behavior policy specification.

In this work, we investigate the use of resampling for online off-policy prediction for known, unchanging target and behavior policies. We first introduce Importance Resampling (IR), which samples transitions from a buffer of (recent) transitions according to IS ratios. These sampled transitions are then used for on-policy updates. We show that IR has the same bias as WIS, and that it can be made unbiased and consistent with the inclusion of a batch correction term—even under a sliding window buffer of experience. We provide additional theoretical results characterizing when we might expect the variance to be lower for IR than IS. We then empirically investigate IR on three microworlds and a racing car simulator, learning from images, highlighting that (a) IR is less sensitive to learning rate than IS and V-trace (IS with clipping) and (b) IR converges more quickly in terms of the number of updates.

2 Background

We consider the problem of learning General Value Functions (GVFs) [Sutton et al., 2011]. The agent interacts in an environment defined by a set of states $\mathcal{S}$, a set of actions $\mathcal{A}$ and Markov transition dynamics, with probability $\mathrm{P}(s'|s, a)$ of transitions to state s' when taking action a in state s. A GVF is defined for policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, cumulant $c : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ and continuation function

[1] There is substantial literature on variance reduction for another area called off-policy policy evaluation, but which estimates only a single number or value for a policy (e.g., see [Thomas and Brunskill, 2016]). The resulting algorithms differ substantially, and are not appropriate for learning the value function.

[2] We explicitly use the term prediction rather than policy evaluation to make it clear that we are not learning value functions for control. Rather, our goal is to learn value functions solely for the sake of prediction.

$\gamma : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$, with $C_{t+1} \stackrel{\text{def}}{=} c(S_t, A_t, S_{t+1})$ and $\gamma_{t+1} \stackrel{\text{def}}{=} \gamma(S_t, A_t, S_{t+1})$ for a (random) transition (S_t, A_t, S_{t+1}). The value for a state $s \in \mathcal{S}$ is

$$V(s) \stackrel{\text{def}}{=} \mathbb{E}_\pi \left[G_t | S_t = s \right] \qquad \text{where } G_t \stackrel{\text{def}}{=} C_{t+1} + \gamma_{t+1} C_{t+2} + \gamma_{t+1}\gamma_{t+2} C_{t+3} + \dots .$$

The operator $\mathbb{E}_\pi$ indicates an expectation with actions selected according to policy π. GVFs encompass standard value functions, where the cumulant is a reward. Otherwise, GVFs enable predictions about discounted sums of others signals into the future, when following a target policy π. These values are typically estimated using parametric function approximation, with weights $\theta \in \mathbb{R}^d$ defining approximate values $V_\theta(s)$.

In off-policy learning, transitions are sampled according to behavior policy, rather than the target policy. To get an unbiased sample of an update to the weights, the action probabilities need to be adjusted. Consider on-policy temporal difference (TD) learning, with update $\alpha_t \delta_t \nabla_\theta V_\theta(s)$ for a given $S_t = s$, for learning rate $\alpha_t \in \mathbb{R}^+$ and TD-error $\delta_t \stackrel{\text{def}}{=} C_{t+1} + \gamma_{t+1} V_\theta(S_{t+1}) - V_\theta(s)$. If actions are instead sampled according to a behavior policy $\mu : \mathcal{S} \times \mathcal{A} \to [0, 1]$, then we can use importance sampling (IS) to modify the update, giving the off-policy TD update $\alpha_t \rho_t \delta_t \nabla_\theta V_\theta(s)$ for IS ratio $\rho_t \stackrel{\text{def}}{=} \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)}$. Given state $S_t = s$, if $\mu(a|s) > 0$ when $\pi(a|s) > 0$, then the expected value of these two updates are equal. To see why, notice that

$$\mathbb{E}_\mu \left[\alpha_t \rho_t \delta_t \nabla_\theta V_\theta(s) | S_t = s \right] = \alpha_t \nabla_\theta V_\theta(s) \mathbb{E}_\mu \left[\rho_t \delta_t | S_t = s \right]$$

which equals $\mathbb{E}_\pi \left[\alpha_t \rho_t \delta_t \nabla_\theta V_\theta(s) | S_t = s \right]$ because

$$\mathbb{E}_\mu \left[\rho_t \delta_t | S_t = s \right] = \sum_{a \in \mathcal{A}} \mu(a|s) \frac{\pi(a|s)}{\mu(a|s)} \mathbb{E} \left[\delta_t | S_t = s, A_t = a \right] = \mathbb{E}_\pi \left[\delta_t | S_t = s \right].$$

Though unbiased, IS can be high-variance. A lower variance alternative is Weighted IS (WIS). For a batch consisting of transitions $\{(s_i, a_i, s_{i+1}, c_{i+1}, \rho_i)\}_{i=1}^n$, batch WIS uses a normalized estimate for the update. For example, an offline batch WIS TD algorithm, denoted WIS-Optimal below, would use update $\alpha_t \frac{\rho_t}{\sum_{i=1}^n \rho_i} \delta_t \nabla_\theta V_\theta(s)$. Obtaining an efficient WIS update is not straightforward, however, when learning online and has resulted in algorithms in the SGD setting (i.e. $n = 1$) specialized to tabular [Precup et al., 2001] and linear functions [Mahmood et al., 2014, Mahmood and Sutton, 2015]. We nonetheless use WIS as a baseline in the experiments and theory.

3 Importance Resampling

In this section, we introduce Importance Resampling (IR) for off-policy prediction and characterize its bias and variance. A resampling strategy requires a buffer of samples, from which we can resample. Replaying experience from a buffer was introduced as a biologically plausible way to reuse old experience [Lin, 1992, 1993], and has become common for improving sample efficiency, particularly for control [Mnih et al., 2015, Schaul et al., 2015b]. In the simplest case—which we assume here—the buffer is a sliding window of the most recent n samples, $\{(s_i, a_i, s_{i+1}, c_{i+1}, \rho_i)\}_{i=t-n}^t$, at time step $t > n$. We assume samples are generated by taking actions according to behavior μ. The transitions are generated with probability $d_\mu(s)\mu(a|s)\mathrm{P}(s'|s, a)$, where $d_\mu : \mathcal{S} \to [0, 1]$ is the stationary distribution for policy μ. The goal is to obtain samples according to $d_\mu(s)\pi(a|s)\mathrm{P}(s'|s, a)$, as if we had taken actions according to policy π from states[3] $s \sim d_\mu$.

The IR algorithm is simple: resample a mini-batch of size k on each step t from the buffer of size n, proportionally to ρ_i in the buffer. Using the resampled mini-batch we can update our value function using standard on-policy approaches, such as on-policy TD or on-policy gradient TD. The key difference to IS and WIS is that the distribution itself is corrected, before the update, whereas IS and WIS correct the update itself. This small difference, however, can have larger ramifications practically, as we show in this paper.

[3]The assumption that states are sampled from d_μ underlies most off-policy learning algorithms. Only a few attempt to adjust probabilities d_μ to d_π, either by multiplying IS ratios before a transition [Precup et al., 2001] or by directly estimating state distributions [Hallak and Mannor, 2017, Liu et al., 2018]. In this work, we focus on using resampling to correct the action distribution—the standard setting. We expect, however, that some insights will extend to how to use resampling to correct the state distribution, particularly because wherever IS ratios are used it should be straightforward to use our resampling approach.

We consider two variants of IR: with and without bias correction. For point i_j sampled from the buffer, let Δ_{i_j} be the on-policy update for that transition. For example, for TD, $\Delta_{i_j} = \delta_{i_j} \nabla_\theta V_\theta(s_{i_j})$. The first step for either variant is to sample a mini-batch of size k from the buffer, proportionally to ρ_i. Bias-Corrected IR (BC-IR) additionally pre-multiplies with the average ratio in the buffer $\bar{\rho} \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \rho_i$, giving the following estimators for the update direction

$$X_{\text{IR}} \stackrel{\text{def}}{=} \frac{1}{k} \sum_{j=1}^{k} \Delta_{i_j} \qquad\qquad X_{\text{BC}} \stackrel{\text{def}}{=} \frac{\bar{\rho}}{k} \sum_{j=1}^{k} \Delta_{i_j}$$

BC-IR negates bias introduced by the average ratio in the buffer deviating significantly from the true mean. For reasonably large buffers, $\bar{\rho}$ will be close to 1 making IR and BC-IR have near-identical updates[4]. Nonetheless, they do have different theoretical properties, particularly for small buffer sizes n, so we characterize both.

Across most results, we make the following assumption.

Assumption 1. *A buffer $B_t = \{X_{t+1}, ..., X_{t+n}\}$ is constructed from the most recent n transitions sampled by time $t+n$, which are generated sequentially from an irreducible, finite MDP with a fixed policy μ.*

To denote expectations under $p(x) = d_\mu(s)\mu(a|s)\text{P}(s'|s,a)$ and $q(x) = d_\mu(s)\pi(a|s)\text{P}(s'|s,a)$, we overload the notation from above, using operators $\mathbb{E}_\mu$ and $\mathbb{E}_\pi$ respectively. To reduce clutter, we write $\mathbb{E}$ to mean $\mathbb{E}_\mu$, because most expectations are under the sampling distribution. All proofs can be found in Appendix B.

3.1 Bias of IR

We first show that IR is biased, and that its bias is actually equal to WIS-Optimal, in Theorem 3.1.

Theorem 3.1. *[Bias for a fixed buffer of size n] Assume a buffer B of n transitions sampled i.i.d according to $p(x = (s,a,s')) = d_\mu(s)\mu(a|s)\text{P}(s'|s,a)$. Let $X_{\text{WIS}^*} \stackrel{\text{def}}{=} \sum_{i=1}^{n} \frac{\rho_i}{\sum_{j=1}^{n} \rho_j} \Delta_i$ be the WIS-Optimal estimator of the update. Then,*

$$\mathbb{E}[X_{\text{IR}}] = \mathbb{E}[X_{\text{WIS}^*}]$$

and so the bias of X_{IR} is proportional to

$$\text{Bias}(X_{\text{IR}}) = \mathbb{E}[X_{\text{IR}}] - \mathbb{E}_\pi[\Delta] \propto \frac{1}{n}(\mathbb{E}_\pi[\Delta]\sigma_\rho^2 - \sigma_{\rho,\Delta}\sigma_\rho\sigma_\Delta) \tag{1}$$

where $\mathbb{E}_\pi[\Delta]$ is the expected update across all transitions, with actions from S taken by the target policy π; $\sigma_\rho^2 = \text{Var}(\frac{1}{n}\sum_{j=1}^{n}\rho_j)$; $\sigma_\Delta^2 = \text{Var}(\frac{1}{n}\sum_{i=1}^{n}\rho_i\Delta_i)$; and covariance $\sigma_{(\rho,\Delta)} = \text{Cov}(\frac{1}{n}\sum_{j=1}^{n}\rho_j, \frac{1}{n}\sum_{i=1}^{n}\rho_i\Delta_i)$.

Theorem 3.1 is the only result which follows a different set of assumptions, primarily due to using the bias characterization of X_{WIS^*} found in Owen [2013]. The bias of IR will be small for reasonably large n, both because it is proportional to $1/n$ and because larger n will result in lower variance of the average ratios and average update for the buffer in Equation (1). In particular, as n grows, these variances decay proportionally to n. Nonetheless, for smaller buffers, such bias could have an impact. We can, however, easily mitigate this bias with a bias-correction term, as shown in the next corollary and proven in Appendix B.2.

Corollary 3.1.1. *BC-IR is unbiased: $\mathbb{E}[X_{\text{BC}}] = \mathbb{E}_\pi[\Delta]$.*

3.2 Consistency of IR

Consistency of IR in terms of an increasing buffer, with $n \to \infty$, is a relatively straightforward extension of prior results for SIR, with or without the bias correction, and from the derived bias of both estimators (see Theorem B.1 in Appendix B.3). More interesting, and reflective of practice, is consistency *with a fixed length buffer* and increasing interactions with the environment, $t \to \infty$. IR, without bias correction, is asymptotically biased in this case; in fact, its asymptotic bias is the one characterized above for a fixed length buffer in Theorem 3.1. BC-IR, on the other hand, is consistent, even with a sliding window, as we show in the following theorem.

[4] $\bar{\rho} \approx \mathbb{E}[\rho(a|s)] = \mathbb{E}[\frac{\pi(a|s)}{\mu(a|s)}] = \sum_{s,a} \frac{\pi(a|s)}{\mu(a|s)}\mu(a|s)d_\mu(s) = 1$.

Theorem 3.2. *Let $B_t = \{X_{t+1}, ..., X_{t+n}\}$ be the buffer of the most recent n transitions sampled according to Assumption 1. Define the sliding-window estimator $X_t \overset{\text{def}}{=} \frac{1}{T}\sum_{t=1}^{T} X_{\text{BC}}^{(t)}$. Then, if $\mathbb{E}_\pi[|\Delta|] < \infty$, then X_T converges to $\mathbb{E}_\pi[\Delta]$ almost surely as $T \to \infty$.*

3.3 Variance of Updates

It might seem that resampling avoids high-variance in updates, because it does not reweight with large magnitude IS ratios. The notion of *effective sample size* from statistics, however, provides some intuition about why large magnitude IS ratios can also negatively affect IR, not just IS. Effective sample size is between 1 and n, with one estimator $\left(\sum_{i=1}^n \rho_i\right)^2 / \sum_{i=1}^n \rho_i^2$ [Kong et al., 1994, Martino et al., 2017]. When the effective sample size is low, this indicates that most of the probability is concentrated on a few samples. For high magnitude ratios, IR will repeatedly sample the same transitions, and potentially never sample some of the transitions with small IS ratios.

Fortunately, we find that, despite this dependence on effective sample size, IR can significantly reduce variance over IS. In this section, we characterize the variance of the BC-IR estimator. We choose this variant of IR, because it is unbiased and so characterizing its variance is a more fair comparison to IS. We define the mini-batch IS estimator $X_{\text{IS}} \overset{\text{def}}{=} \frac{1}{k}\sum_{j=1}^{k} \rho_{z_j}\Delta_{z_j}$, where indices z_j are sampled uniformly from $\{1, \ldots, n\}$. This contrasts the indices $i_1, \ldots, i_k$ for X_{BC} that are sampled proportionally to ρ_i.

We begin by characterizing the variance, under a fixed dataset B. For convenience, let $\mu_B = \mathbb{E}_\pi[\Delta|B]$. We characterize the sum of the variances of each component in the update estimator, which equivalently corresponds to normed deviation of the update from its mean,

$$\mathbb{V}(\Delta \mid B) \overset{\text{def}}{=} \operatorname{tr}\operatorname{Cov}(\Delta \mid B) = \sum_{m=1}^{d} \operatorname{Var}(\Delta_m \mid B) = \mathbb{E}[\|\Delta - \mu_B\|_2^2 \mid B]$$

for an unbiased stochastic update $\Delta \in \mathbb{R}^d$. We show two theorems that BC-IR has lower variance than IS, with two different conditions on the norm of the update. We first start with more general conditions, and then provide a theorem for conditions that are likely only true in early learning.

Theorem 3.3. *Assume that, for a given buffer B, $\|\Delta_j\|_2^2 > \frac{c}{\rho_j}$ for samples where $\rho_j \geq \bar{\rho}$, and that $\|\Delta_j\|_2^2 < \frac{c}{\rho_j}$ for samples where $\rho_j < \bar{\rho}$, for some $c > 0$. Then the BC-IR estimator has lower variance than the IS estimator: $\mathbb{V}(X_{\text{BC}} \mid B) < \mathbb{V}(X_{\text{IS}} \mid B)$.*

The conditions in Theorem 3.3 preclude having update norms for samples with small ρ be quite large—larger than a number $\propto \frac{1}{\rho}$—and a small norm for samples with large ρ. These conditions can be relaxed to a statement on average, where the cumulative weighted magnitude of the update norm for samples with ρ below the median needs to be smaller than for samples with ρ above the mean (see the proof in Appendix B.5).

We next consider a setting where the magnitude of the update is independent of the given state and action. We expect this condition to hold in early learning, where the weights are randomly initialized, and thus randomly incorrect across the state-action space. As learning progresses, and value estimates become more accurate in some states, it is unlikely for this condition to hold.

Theorem 3.4. *Assume ρ and the magnitude of the update $\|\Delta\|_2^2$ are independent*

$$\mathbb{E}[\rho_j \|\Delta_j\|_2^2 \mid B] = \mathbb{E}[\rho_j \mid B]\, \mathbb{E}[\|\Delta_j\|_2^2 \mid B]$$

Then the BC-IR estimator will have equal or lower variance than the IS estimator: $\mathbb{V}(X_{\text{BC}} \mid B) \leq \mathbb{V}(X_{\text{IS}} \mid B)$.

These results have focused on variance of each estimator, for a fixed buffer, which provided insight into variance of updates when executing the algorithms. We would, however, also like to characterize variability across buffers, especially for smaller buffers. Fortunately, such a characterization is a simple extension on the above results, because variability for a given buffer already demonstrates variability due to different samples. It is easy to check that $\mathbb{E}[\mathbb{E}[\mu_{IR} \mid B]] = \mathbb{E}[\mu_{IS} \mid B] = \mathbb{E}_\pi[\Delta]$. The variances can be written using the law of total variance

$$\begin{aligned}
\mathbb{V}(X_{\text{BC}}) &= \mathbb{E}[\mathbb{V}(X_{\text{BC}} \mid B)] + \mathbb{V}(\mathbb{E}[X_{\text{BC}} \mid B]) = \mathbb{E}[\mathbb{V}(X_{\text{BC}} \mid B)] + \mathbb{V}(\mu_B) \\
\mathbb{V}(X_{\text{IS}}) &= \mathbb{E}[\mathbb{V}(X_{\text{IS}} \mid B)] + \mathbb{V}(\mu_B) \\
\implies \mathbb{V}(X_{\text{BC}}) - \mathbb{V}(X_{\text{IS}}) &= \mathbb{E}[\mathbb{V}(X_{\text{BC}} \mid B) - \mathbb{V}(X_{\text{IS}} \mid B)]
\end{aligned}$$

with expectation across buffers. Therefore, the analysis of $\mathbb{V}(X_{\text{BC}} \mid B)$ directly applies.

4 Empirical Results

We investigate the two hypothesized benefits of resampling as compared to reweighting: improved sample efficiency and reduced variance. These benefits are tested in two microworld domains—a Markov chain and the Four Rooms domain—where exhaustive experiments can be conducted. We also provide a demonstration that IR reduces sensitivity over IS and VTrace in a car simulator, TORCs, when learning from images [5].

We compare IR and BC-IR against several reweighting strategies, including importance sampling (IS); two online approaches to weighted important sampling, WIS-Minibatch with weighting $\rho_i / \sum_{j=1}^{k} \rho_j$ and WIS-Buffer with weighting $\rho_i / \frac{k}{n} \sum_{j=1}^{n} \rho_j$; and V-trace[6], which corresponds to clipping importance weights [Espeholt et al., 2018]. We also compare to WIS-TD(0) [Mahmood and Sutton, 2015], when applicable, which uses an online approximation to WIS, with a stepsize selection strategy (as described in Appendix A.2). This algorithm uses only one sample at a time, rather than a mini-batch, and so is only included in Figure 2. Where appropriate, we also include baselines using On-policy sampling; WIS-Optimal which uses the whole buffer to get an update; and Sarsa(0) which learns action-values—which does not require IS ratios—and then produces estimate $V(s) = \sum_a \pi(s,a) Q(s,a)$. WIS-Optimal is included as an optimal baseline, rather than as a competitor, as it estimates the update using the whole buffer on every step.

In all the experiments, the data is generated off-policy. We compute the absolute value error (AVE) or the absolute return error (ARE) on every step. For the sensitivity plots we take the average over all the interactions as specified for the environment — resulting in MAVE and MARE respectively. The error bars represent the standard error over runs, which are featured on every plot — although not visible in some instances. For the microworlds, the true value function is found using dynamic programming with threshold 10^{-15}, and we compute AVE over all the states. For TORCs and continuous Four Rooms, the true value function is approximated using rollouts from a random subset of states generated when running the behavior policy μ, and the ARE is computed over this subset. For the Torcs domain, the same subset of states is used for each run due to computational constraints and report the mean squared return error (MSRE). Plots showing sensitivity over number of updates show results for complete experiments with updates evenly spread over all the interactions. A tabular representation is used in the microworld experiments, tilecoded features with 64 tilings and 8 tiles is used in continuous Four Rooms, and a convolutional neural network is used for TORCs, with an architecture previously defined for self-driving cars [Bojarski et al., 2016].

4.1 Investigating Convergence Rate

We first investigate the convergence rate of IR. We report learning curves in Four Rooms, as well as sensitivity to the learning rate. The Four Rooms domain [Stolle and Precup, 2002] has four rooms in an 11x11 grid world. The four rooms are positioned in a grid pattern with each room having two adjacent rooms. Each adjacent room is separated by a wall with a single connecting hallway. The target policy takes the down action deterministically. The cumulant for the value function is 1 when the agent hits a wall and 0 otherwise. The continuation function is $\gamma = 0.9$, with termination when the agent hits a wall. The resulting value function can be thought of as distance to the bottom wall. The behavior policy is uniform random everywhere except for 25 randomly selected states which take the action down with probability 0.05 with remaining probability split equally amongst the other actions. The choice of behavior and target policy induce high magnitude IS ratios.

As shown in Figure 1, IR has noticeable improvements over the reweighting strategies tested. The fact that IR resamples more important transitions from the replay buffer seems to significantly increase the learning speed. Further, IR has a wider range of usable learning rates. The same effect is seen even as we reduce the total number of updates, where the uniform sampling methods perform significantly worse as the interactions between updates increases—suggesting improved sample efficiency. WIS-Buffer performs almost equivalently to IS, because for reasonably size buffers, its normalization factor $\frac{1}{n} \sum_{j=1}^{n} \rho_j \approx 1$ because $\mathbb{E}[\rho] = 1$. WIS-Minibatch and V-trace both reduce

[5] Experimental code for every domain except Torcs can be found at `https://mkschleg.github.io/Resampling.jl`

[6] Retrace, ABQ and TreeBackup also use clipping to reduce variance. But, they are designed for learning action-values and for mitigating variance in eligibility traces. When trace parameter $\lambda = 0$—as we assume here—there are no IS ratios and these methods become equivalent to using Sarsa(0) for learning action-values.

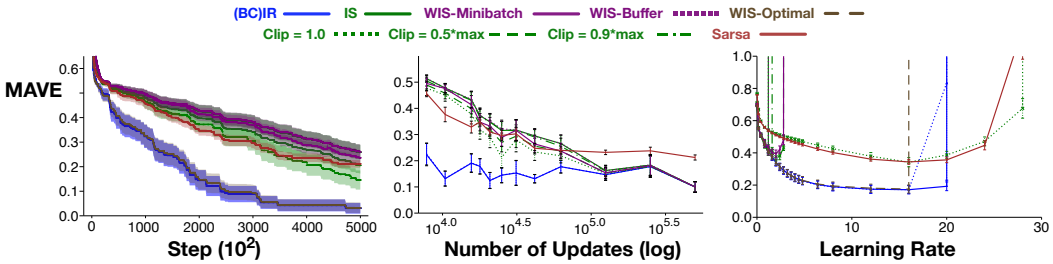

Figure 1: Four Rooms experiments ($n = 2500$, $k = 16$, 25 runs): **left** Learning curves for each method, with updates every 16 steps. IR and WIS-Optimal are overlapping. **center** Sensitivity over the number of interactions between updates. **right** Learning rate sensitivity plot.

the variance significantly, with their bias having only a limited impact on the final performance compared to IS. Even the most aggressive clipping parameter for V-trace—a clipping of 1.0— outperforms IS. The bias may have limited impact because the target policy is deterministic, and so only updates for exactly one action in a state. Sarsa—which is the same as Retrace(0)—performs similarly to the reweighting strategies.

The above results highlight the convergence rate improvements from IR, in terms of number of updates, without generalization across values. Conclusions might actually be different with function approximation, when updates for one state can be informative for others. For example, even if in one state the target policy differs significantly from the behavior policy, if they are similar in a related state, generalization could overcome effective sample size issues. We therefore further investigate if the above phenomena arise under function approximation with RMSProp learning rate selection.

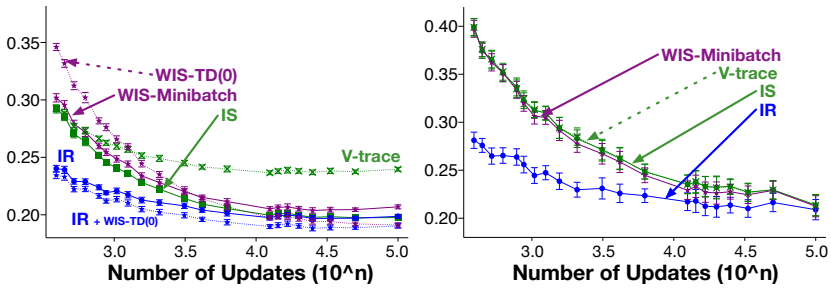

Figure 2: Convergence rates in Continuous Four Rooms averaged over 25 runs with 100000 interactions with the environment. **left** uniform random behavior policy and target policy which takes the down action with probability 0.9 and probability $0.1/3$ for all other actions. Learning used incremental updates (as specified in appendix A.2). **right** uniform random behavior and target policy with persistent down action selection learned with mini-batch updates with RMSProp.

We conduct two experiments similar to above, in a continuous state Four Rooms variant. The agent is a circle with radius 0.1, and the state consists of a continuous tuple containing the x and y coordinates of the agent's center point. The agent takes an action in one of the 4 cardinal directions moving $0.5 \pm \mathcal{U}(0.0, 0.1)$ in that directions with random drift in the orthogonal direction sampled from $\mathcal{N}(0.0, 0.01)$. The representation is a tile coded feature vector with 64 tilings and 8 tiles. We provide results for both mini-batch updating (as above) and incremental updating (i.e. updating on each transition of a mini-batch incrementally, see appendix A.2 for details). For the mini-batch experiment, the target policy deterministically takes the down action. For the incremental experiment, the target policy takes the down action with probability 0.9 and selects all other action with probability $0.1/3$.

We find that generalization can mitigate some of the differences between IR and IS above in some settings, but in others the difference remains just as stark (see Figure 2 and Appendix C.2). If we use the behavior policy from the tabular domain, which skews the behavior in a sparse set of states, the nearby states mitigate this skew. However, if we use a behavior policy that selects all actions

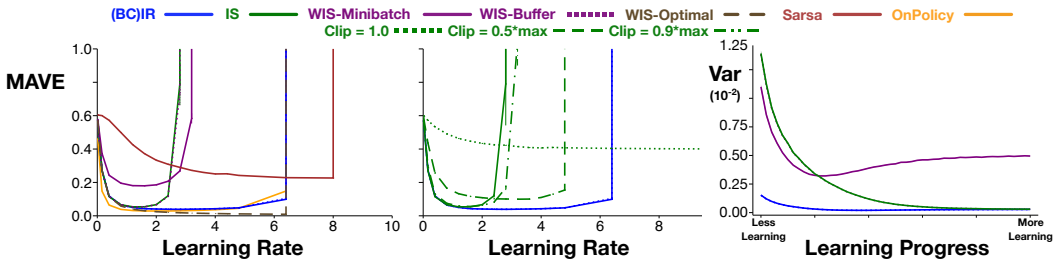

Figure 4: Learning Rate sensitivity plots in the Random Walk Markov Chain, with buffer size $n = 15000$ and mini-batch size $k = 16$. Averaged over 100 runs. The policies, written as [probability left, probability right] are $\mu = [0.9, 0.1], \pi = [0.1, 0.9]$ **left** learning rate sensitivity plot for all methods but V-trace. **center** learning rate sensitivity for V-trace with various clipping parameters **right** Variance study for IS, IR, and WISBatch. The x-axis corresponds to the training iteration, with variance reported for the weights at that iteration generated by WIS-Optimal. These plots show a correlation between the sensitivity to learning rate and magnitude of variance.

uniformly, then again IR obtains noticeable gains over IS and V-trace, for reducing the required number of updates, as shown in Figure 2.

We find similar results for the incremental setting Figure 2 (left), where resampling still outperforms all other methods in terms of convergence rates. Given WIS-TD(0)'s significant degrade in performance as the number of updates decreases, we also compare with using WIS-TD(0) when sampling according to resampling IR+WIS-TD(0). Interestingly, this method outperforms all others — albeit only slightly against IR with constant learning rate. This result leads us to believe RMSProp may be a optimizer poor choice for this setting. Expanded results can be found in Appendix C.2.

4.2 Investigating Variance

To better investigate the update variance we use a Markov chain, where we can more easily control dissimilarity between μ and π, and so control the magnitude of the IS ratios. The Markov chain is composed of 8 non-terminating states and 2 terminating states on the ends of the chain, with a cumulant of 1 on the transition to the right-most terminal state and 0 everywhere else. We consider policies with probabilities [left, right] equal in all states: $\mu = [0.9, 0.1], \pi = [0.1, 0.9]$; further policy settings can be found in Appendix C.1.

We first measure the variance of the updates for fixed buffers. We compute the variance of the update—from a given weight vector—by simulating the many possible updates that could have occurred. We are interested in the variance of updates both for early learning—when the weight vector is quite incorrect and updates are larger—and later learning. To obtain a sequence of such weight vectors, we use the sequence of weights generated by WIS-Optimal. As shown in Figure 4, the variance of IR is lower than IS, particularly in early learning, where the difference is stark. Once the weight vector has largely converged, the variance of IR and IS is comparable and near zero.

We can also evaluate the update variance by proxy using learning rate sensitivity curves. As seen in Figure 4 (left) and (center), IR has the lowest sensitivity to learning rates, on-par with On-Policy sampling. IS has the highest sensitivity, along with WIS-Buffer and WIS-Minibatch. Various clipping parameters with V-trace are also tested. V-trace does provide some level of variance reduction but incurs more bias as the clipping becomes more aggressive.

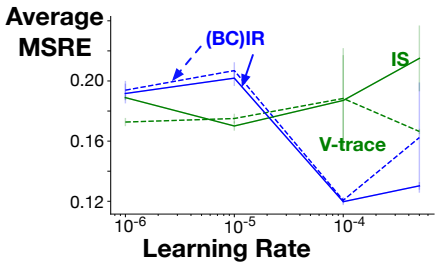

Figure 3: Learning rate sensitivity in TORCs, averaged over 10 runs. V-trace has clipping parameter 1.0. All the methods performed worse with a higher learning rate than shown here, so we restrict to this range.

4.3 Demonstration on a Car Simulator

We use the TORCs racing car simulator to perform scaling experiments with neural networks to compare IR, IS, and V-trace. The simulator produces

64x128 cropped grayscale images. We use an underlying deterministic steering controller that produces steering actions $a_{det} \in [-1, +1]$ and take an action with probability defined by a Gaussian $a \sim \mathcal{N}(a_{det}, 0.1)$. The target policy is a Gaussian $\mathcal{N}(0.15, 0.0075)$, which corresponds to steering left. Pseudo-termination (i.e., $\gamma = 0$) occurs when the car nears the center of the road, and the cumulant becomes 1. Otherwise, the cumulant is zero and $\gamma = 0.9$. The policy is specified using continuous action distributions and results in IS ratios as high as ~ 1000 and highly variant updates for IS.

Again, we can see that IR provides benefits over IS and V-trace, in Figure 3. There is even more generalization from the neural network in this domain, than in Four Rooms where we found generalization did reduce some of the differences between IR and IS. Yet, IR still obtains the best performance, and avoids some of the variance seen in IS for two of the learning rates. Additionally, BC-IR actually performs differently here, having worse performance for the largest learning rate. This suggest IR has an effect in reducing variance.

5 Conclusion

In this paper we introduced a new approach to off-policy learning: Importance Resampling. We showed that IR is consistent, and that the bias is the same as for Weighted Importance Sampling. We also provided an unbiased variant of IR, called Bias-Corrected IR. We empirically showed that IR (a) has lower learning rate sensitivity than IS and V-trace, which is IS with varying clipping thresholds; (b) the variance of updates for IR are much lower in early learning than IS and (c) IR converges faster than IS and other competitors, in terms of the number of updates. These results confirm the theory presented for IR, which states that variance of updates for IR are lower than IS in two settings, one being an early learning setting. Such lower variance also explains why IR can converge faster in terms of number of updates, for a given buffer of data.

The algorithm and results in this paper suggest new directions for off-policy prediction, particularly for faster convergence. Resampling is promising for scaling to learning many value functions in parallel, because many fewer updates can be made for each value function. A natural next step is a demonstration of IR, in such a parallel prediction system. Resampling from a buffer also opens up questions about how to further focus updates. One such option is using an intermediate sampling policy. Another option is including prioritization based on error, such as was done for control with prioritized sweeping [Peng and Williams, 1993] and prioritized replay [Schaul et al., 2015b].

Acknowledgments

We would like to thank Huawei for their support, and especially for allowing a portion of this work to be completed during Matthews internship in the summer of 2018. We also would like to acknowledge University of Alberta, Alberta Machine Intelligence Institute, IVADO, and NSERC for their continued funding and support, as well as Compute Canada (`www.computecanada.ca`) for the computing resources used for this work.

References

Sigrun Andradottir, Daniel P Heyman, and Teunis J Ott. On the Choice of Alternative Measures in Importance Sampling with Markov Chains. *Operations Research*, 1995.

M S Arulampalam, S Maskell, N Gordon, and T Clapp. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, 2002.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Lasse Espeholt, Hubert Soyer, Rémi Munos, Karen Simonyan, Volodymir Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, and others. IMPALA: Scalable distributed Deep-RL with importance weighted actor-learner architectures. *arXiv preprint arXiv:1802.01561*, 2018.

N J Gordon, D J Salmond, Radar, AFM Smith IEE Proceedings F Signal, and 1993. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IET*, 1993.

Assaf Hallak and Shie Mannor. Consistent on-line off-policy evaluation. *arXiv preprint arXiv:1702.07121*, 2017.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement Learning with Unsupervised Auxiliary Tasks. In *International Conference on Representation Learning*, 2017.

H Kahn and A W Marshall. Methods of Reducing Sample Size in Monte Carlo Computations. *Journal of the Operations Research Society of America*, 1953.

Augustine Kong, Jun S Liu, and Wing Hung Wong. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 1994.

David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

Long-Ji Lin. Self-Improving Reactive Agents Based On Reinforcement Learning, Planning and Teaching. *Machine Learning*, 1992.

Long-Ji Lin. *Reinforcement Learning for Robots Using Neural Networks*. PhD thesis, Carnegie Mellon University, 1993.

Qiang Liu, Lihong Li, Ziyang Tang, and Dengyong Zhou. Breaking the Curse of Horizon: Infinite-Horizon Off-Policy Estimation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018.

Victoria Lopez, Alberto Fernandez, Salvador Garcia, Vasile Palade, and Francisco Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 2013.

A R Mahmood and R.S. Sutton. Off-policy learning based on weighted importance sampling with linear computational complexity. In *Conference on Uncertainty in Artificial Intelligence*, 2015.

A Rupam Mahmood, Hado P van Hasselt, and Richard S Sutton. Weighted importance sampling for off-policy learning with linear function approximation. In *Advances in Neural Information Processing Systems*, 2014.

Ashique Rupam Mahmood, Huizhen Yu, and Richard S Sutton. Multi-step Off-policy Learning Without Importance Sampling Ratios. *arXiv:1509.01240v2*, 2017.

Luca Martino, Víctor Elvira, and Francisco Louzada. Effective sample size for importance sampling based on discrepancy measures. *Signal Processing*, 2017.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 2015.

Joseph Modayil, Adam White, and Richard S Sutton. Multi-timescale nexting in a reinforcement learning robot. *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, 2014.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc G Bellemare. Safe and Efficient Off-Policy Reinforcement Learning. *Advances in Neural Information Processing Systems*, 2016.

Art B. Owen. *Monte Carlo theory, methods and examples*. 2013.

Jing Peng and Ronald J Williams. Efficient Learning and Planning Within the Dyna Framework. *Adaptive Behavior*, 1993.

Doina Precup, Richard S Sutton, and Satinder P Singh. Eligibility Traces for Off-Policy Policy Evaluation. *ICML*, 2000.

Doina Precup, Richard S Sutton, and Sanjoy Dasgupta. Off-Policy Temporal-Difference Learning with Function Approximation. *ICML*, 2001.

Donald B Rubin. Using the SIR algorithm to simulate posterior distributions. *Bayesian statistics*, 1988.

Reuven Y Rubinstein and Dirk P Kroese. *Simulation and the Monte Carlo Method*. John Wiley & Sons, 2016.

Tom Schaul and Mark Ring. Better generalization with forecasts. In *International Joint Conference on Artificial Intelligence*, 2013.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal Value Function Approximators. In *International Conference on Machine Learning*, 2015a.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized Experience Replay. *arXiv:1511.05952 [cs]*, 2015b.

David Silver, Hado van Hasselt, Matteo Hessel, Tom Schaul, Arthur Guez, Tim Harley, Gabriel Dulac-Arnold, David P Reichert, Neil C Rabinowitz, André Barreto, and Thomas Degris. The Predictron - End-To-End Learning and Planning. In *AAAI Conference on Artificial Intelligence*, 2017.

Øivind Skare, Erik Bølviken, and Lars Holden. Improved Sampling-Importance Resampling and Reduced Bias Importance Sampling. *Scandinavian Journal of Statistics*, 2003.

AFM Smith, AE Gelfand The American Statistician, and 1992. Bayesian statistics without tears: a sampling–resampling perspective. *Taylor & Francis*, 1992.

Martin Stolle and Doina Precup. Learning Options in Reinforcement Learning. In *International Symposium on Abstraction, Reformulation, and Approximation*, 2002.

Richard S Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 1999.

Richard S Sutton, Eddie J Rafols, and Anna Koop. Temporal Abstraction in Temporal-difference Networks. In *Advances in Neural Information Processing Systems*, 2005.

Richard S Sutton, J Modayil, M Delp, T Degris, P.M. Pilarski, A White, and D Precup. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *International Conference on Autonomous Agents and Multiagent Systems*, 2011.

Philip Thomas and Emma Brunskill. Data-Efficient Off-Policy Policy Evaluation for Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*, 2016.

Philip S Thomas and Emma Brunskill. Importance Sampling with Unequal Support. In *AAAI Conference on Artificial Intelligence*, 2017.

A Weighted Importance Sampling

A.1 Mini-Batch Algorithms

We consider three weighted importance sampling updates as competitors to IR. n is the size of the experience replay buffer, k is the size of a single batch. WIS-Minibatch and WIS-Buffer both follow a similar protocol as IS, in that they uniformly sample a mini-batch from the experience replay buffer and use this to update the value functions. The difference comes in the scaling of the update. The first, WIS-Minibatch, uses the sum of the importance weights ρ_i in the sampled mini-batch, while WIS-Buffer uses the sum of importance weights in the experience replay buffer. WIS-Buffer is also scaled by the size of the buffer and brought to the same effective scale as the other updates with $\frac{1}{k}$. WIS-Optimal follows a different approach and performs the best possible version of WIS where the gradient descent update is calculated from the whole experience replay buffer. We do not provide analysis on the bias or consistency of WIS-Minibatch or WIS-Buffer, but are natural versions of WIS one might try.

$$\Delta\theta = \frac{\sum_i^k \rho_i \delta_i \nabla_\theta V(s_i;\theta)}{\sum_j^k \rho_j} \qquad \text{WIS-Minibatch}$$

$$\Delta\theta = \frac{n}{k}\frac{\sum_i^k \rho_i \delta_i \nabla_\theta V(s_i;\theta)}{\sum_j^n \rho_j} \qquad \text{WIS-Buffer}$$

$$\Delta\theta = \frac{\sum_i^n \rho_i \delta_i \nabla_\theta V(s_i;\theta)}{\sum_j^n \rho_j} \qquad \text{WIS-Optimal}$$

A.2 Incremental Algorithm

While implementing an efficient true WIS algorithm for mini-batch updating is beyond the scope of this work, we compare WIS-TD(0) to the incremental versions of IR, IS, VTrace, and WISBatch. The difference between the mini-batch and incremental algorithms is how the updates are calculated. In the incremental scheme a random mini-batch of data is similarly sampled from the buffer. We then use each sample individually to update our value function. We do this to more naturally compare our baselines to WIS-TD(0) Mahmood and Sutton [2015]. WIS-TD(0) has parameters $u_0 \in \{\frac{1}{64}, 1, 5, 10, 50\} * 64, \mu \in 10^{-2:0.25:1}$, and $\eta = \frac{\mu}{u_0}$. WIS-TD(0) follows the update equations:

$$\begin{aligned}
\mathbf{u}_{i+1} &= (\mathbf{1} - \eta\boldsymbol{\phi}_i \circ \boldsymbol{\phi}_i) \circ \mathbf{u}_i + \rho_i \boldsymbol{\phi}_i \circ \boldsymbol{\phi}_i \quad \triangleright \circ \stackrel{\text{def}}{=} \text{ element wise product} \\
\alpha_{i+1} &= \mathbf{1} \oslash \mathbf{u}_{t+1} \quad \triangleright \oslash \stackrel{\text{def}}{=} \text{ element wise division} \\
\bar{\delta}_i &= C_i + \gamma_i \boldsymbol{\theta}_i^\top \boldsymbol{\phi}_i' - \boldsymbol{\theta}_{i-1}^\top \boldsymbol{\phi}_i \\
\boldsymbol{\theta}_{i+1} &= \boldsymbol{\theta}_t + \boldsymbol{\alpha}_{i+1} \circ \rho_i (\boldsymbol{\theta}_{i-1}^\top \boldsymbol{\phi}_i - \boldsymbol{\theta}_i^\top \boldsymbol{\phi}_i)\boldsymbol{\phi}_i + \rho_i \bar{\delta}_i \boldsymbol{\alpha}_{i+1} \circ \boldsymbol{\phi}_i
\end{aligned}$$

where $\boldsymbol{\theta} \in \mathbb{R}^d$ is the weight vector of the value function, $\boldsymbol{\phi}_i \in \mathbb{R}^d$ is the feature vector of the i-th transition in the experience replay buffer, and $\boldsymbol{\phi}_i' \in \mathbb{R}^d$ is the feature vector of the next state of the i-th transition in the experience replay buffer.

B Additional Theoretical Results and Proofs

B.1 Bias of IR

Theorem 3.1(Bias for a fixed buffer of size n) Assume a buffer B of n transitions sampled i.i.d according to $p(x = (s, a, s')) = d_\mu(s)\mu(a|s)\mathrm{P}(s'|s, a)$. Let $X_{\mathrm{WIS}^*} \stackrel{\text{def}}{=} \sum_{i=1}^n \frac{\rho_i}{\sum_{j=1}^n \rho_j}\Delta_i$ be the WIS-Optimal estimator of the update. Then,

$$\mathbb{E}[X_{\mathrm{IR}}] = \mathbb{E}[X_{\mathrm{WIS}^*}]$$

and so the bias of X_{IR} is proportional to

$$\text{Bias}(X_{\text{IR}}) = \mathbb{E}[X_{\text{IR}}] - \mathbb{E}_\pi[\Delta] \propto \frac{1}{n}(\mathbb{E}_\pi[\Delta]\sigma_\rho^2 - \sigma_{\rho,\Delta}\sigma_\rho\sigma_\Delta)$$

where $\mathbb{E}_\pi[\Delta]$ is the expected update across all transitions, with actions from S taken by the target policy π; $\sigma_\rho^2 = \text{Var}(\frac{1}{n}\sum_{j=1}^n \rho_j)$; $\sigma_\Delta^2 = \text{Var}(\frac{1}{n}\sum_{i=1}^n \rho_i\Delta_i)$; and covariance $\sigma_{(\rho,\Delta)} = \text{Cov}(\frac{1}{n}\sum_{j=1}^n \rho_j, \frac{1}{n}\sum_{i=1}^n \rho_i\Delta_i)$.

Proof. Notice first that when we weight with ρ_i, this is equivalent to weighting with $\frac{d_\mu(S_i)\pi(A_i|S_i)\text{P}(S_{i+1}|S_i,A_i)}{d_\mu(S_i)\mu(A_i|S_i)\text{P}(S_{i+1}|S_i,A_i)}$, and so is the correct IS ratio for the transition.

$$\begin{aligned}
\mathbb{E}[X_{\text{IR}}] &= \mathbb{E}\left[\mathbb{E}[X_{\text{IR}}|B]\right] = \mathbb{E}\Big[\mathbb{E}\Big[\frac{1}{k}\sum_{j=1}^k \Delta_{i_j}|B\Big]\Big] \\
&= \mathbb{E}\Big[\frac{1}{k}\sum_{j=1}^k \mathbb{E}[\Delta_{i_j}|B]\Big] \qquad \triangleright\, \mathbb{E}[\Delta_{i_j}|B] = \sum_{i=1}^n \frac{\rho_i}{\sum_{j=1}^n \rho_j}\Delta_i \\
&= \mathbb{E}\Big[\sum_{i=1}^n \frac{\rho_i}{\sum_{j=1}^n \rho_j}\Delta_i\Big] \\
&= \mathbb{E}[X_{\text{WIS}^*}].
\end{aligned}$$

This bias of X_{IR} is the same as X_{WIS^*}, which is characterized in Owen [2013], completing the proof. □

B.2 Proof of Unbiasedness of BC-IR

Corollary 3.1.1 BC-IR is unbiased: $\mathbb{E}[X_{\text{BC}}] = \mathbb{E}_\pi[\Delta]$.

Proof.

$$\begin{aligned}
\mathbb{E}[X_{\text{BC}}] &= \mathbb{E}\Big[\frac{\bar{\rho}}{k}\sum_{j=1}^k \mathbb{E}[\Delta_{i_j}|B]\Big] = \mathbb{E}\Big[\bar{\rho}\sum_{i=1}^n \frac{\rho_i}{\sum_{j=1}^n \rho_j}\Delta_i\Big] \\
&= \mathbb{E}\Big[\frac{1}{n}\sum_{i=1}^n \rho_i\Delta_i\Big] = \frac{1}{n}\sum_{i=1}^n \mathbb{E}\Big[\frac{\pi(A_i|S_i)}{\mu(A_i|S_i)}\Delta_i\Big] \\
&= \frac{1}{n}\sum_{i=1}^n \mathbb{E}\Big[\frac{d_\mu(S_i)\pi(A_i|S_i)\text{P}(S_{i+1}|S_i,A_i)}{d_\mu(S_i)\mu(A_i|S_i)\text{P}(S_{i+1}|S_i,A_i)}\Delta_i\Big] \\
&= \frac{1}{n}\sum_{i=1}^n \mathbb{E}_\pi\left[\Delta_i\right] = \mathbb{E}_\pi\left[\Delta\right].
\end{aligned}$$

The last equality follows from the fact that the samples are identically distributed. □

B.3 Consistency of the resampling distribution with a growing buffer

We show that the distribution when following a resampling strategy is consistent: as $n \to \infty$, the resampling distribution converges to the true distribution. Our approach closely follows that of [Smith et al., 1992], but we recreate it here for convenience.

Proposition B.1. *Let $X_n = \{x_1, x_2, ..., x_n\}$ be a buffer of data sampled i.i.d. according to proposal density $p(x_i)$. Let $q(x_i)$ be some distribution of interest with associated random variable Q and assume the proposal distribution samples everywhere where $q(\cdot)$ is non-zero, i.e $supp(q) \subseteq supp(p)$. Also, let Y be a discrete random variable taking values x_i with probability $\propto \frac{q(x_i)}{p(x_i)}$. Then, Y converges in distribution to Q as $n \to \infty$.*

Proof. Let $\rho_i = \frac{q(x_i)}{p(x_i)}$. From the probability mass function of Y, we have that:

$$\begin{aligned}
\mathbb{P}[Y \leq a] &= \sum_{i=1}^{n} \mathbb{P}[Y = x_i] \mathbb{1}\{x_i \leq a\} \\
&= \frac{n^{-1} \sum_{i=1}^{n} \rho_i \mathbb{1}\{x_i \leq a\}}{n^{-1} \sum_{i=1}^{n} \rho_i} \\
&\xrightarrow{n \to \infty} \frac{\mathbb{E}_q[\rho(x) \mathbb{1}\{x \leq a\}]}{\mathbb{E}_q[\rho(x)]} \\
&= \frac{1 \cdot \int_{-\infty}^{a} \frac{q(x)}{p(x)} p(x) dx + 0 \cdot \int_{a}^{\infty} \frac{q(x)}{p(x)} p(x) dx}{\int_{-\infty}^{\infty} \frac{q(x)}{p(x)} p(x) dx} \\
&= \int_{-\infty}^{a} q(x) dx = \mathbb{P}[Q \leq a] \\
Y &\xrightarrow{d} Q
\end{aligned}$$

□

This means a resampling strategy effectively changes the distribution of random variable X_n to that of $q(x)$, meaning we can use samples from Y to build statistics about the target distribution $q(x)$. This result motivates using resampling to correct the action distribution in off-policy learning. This result can also be used to show that the IR estimators are consistent, with $n \to \infty$.

B.4 Consistency under a sliding window

Lemma B.2. *Let $B_t = \{X_{t+1}, ..., X_{t+n}\}$ be the buffer of the most recent n transitions sampled by time $t + n$, which are generated sequentially from an irreducible, finite MDP with a fixed policy μ. We define $X_{\mathrm{BC}}^{(t)}$ be the BCIR estimator for buffer B_t. If $\mathbb{E}_\pi[|\Delta|] < \infty$, then $\mathbb{E}[X_{\mathrm{BC}}^{(t)}] = \mathbb{E}_\pi[\Delta]$.*

Proof. Let $X_t = (S_t, A_t, R_{t+1}, S_{t+1})$ be a transition and $\{B_t\}_{t \in \mathbb{N}}$ be the sequence of buffers that are observed, each containing n consecutive transitions.

Using the law of iterated expectations,

$$\mathbb{E}\left[X_{\mathrm{BC}}^{(t)}\right] = \mathbb{E}\left[\mathbb{E}[X_{\mathrm{BC}}^{(t)} | B_t]\right]$$

where the outer expectation is over the stationary distribution of B_t and the inner expectation is over the sampling distribution of IR from the buffer B_t.

Using the definition of $X_{\mathrm{BC}}^{(t)} | B_t$, we have that

$$\begin{aligned}
\mathbb{E}[X_{\mathrm{BC}}^{(t)} | B_t] &= \bar{\rho} \sum_{i=1}^{n} \Delta_i \frac{\rho_i}{\sum_{i=1}^{n} \rho_i} \\
&= \frac{1}{n} \sum_{i=1}^{n} \rho_i \Delta_i
\end{aligned}$$

Next, the stationary distribution of B_t is given by $d(B_t) = Pr(B_t = (x_{t+1}, ..., x_{t+n})) = d_X(x_t) p(x_{t+1}|x_t) ... p(x_{t+n}|x_{t+n-1})$, where d_X is the stationary distribution of X_t. We can verify directly by checking that for all $B' = (x_2, ..., x_{n+1})$

$$\sum_{B} d(B) p(B'|B) = d(B')$$

where $B = (x_1, ..., x_n)$

To see this, first note that $p(B'|B) = p(x_{n+1}|x_n) \mathbf{1}(B, B')$ where $\mathbf{1}(B, B')$ is equal to 1 if the states $(x_2, ..., x_n)$ in B match the states $(x_2, .., x_n)$ in B'. In other words, the first $n - 1$ transitions in B'

must match the last $n-1$ transitions in B for $p(B'|B)$ to be positive. Next, fixing B',

$$\begin{aligned}
&\sum_B d(B)p(B'|B) \\
&= \sum_{x_1,\ldots,x_n} d_\mu(x_1)p(x_2|x_1)...p(x_n|x_{n-1})p(x_{n+1}|x_n)\mathbf{1}(B,B') \\
&= \sum_{x_1} d_\mu(x_1)p(x_2|x_1)...p(x_n|x_{n-1})p(x_{n+1}|x_n) \quad \text{since } (x_2,...,x_n) \text{ have to match} \\
&= d_\mu(x_2)p(x_3|x_2)...p(x_n|x_{n-1})p(x_{n+1}|x_n) \\
&= d(B')
\end{aligned}$$

which verifies the expression for the stationary distribution of B_t.

Continuing from before, we expand the expectation as:

$$\begin{aligned}
&\mathbb{E}\left[\frac{1}{n}\sum_{t=1}^n \rho_t\Delta_t\right] \\
&= \sum_{x_1,\ldots x_n} d_X(x_1)p(x_2|x_1)...p(x_n|x_{n-1})\left(\frac{1}{n}\sum_{t=1}^n \rho_t\Delta_t\right) \\
&= \sum_{\substack{s_1,a_1,r_2,s_2,\ldots,\\ s_n,a_n,r_{n+1},s_{n+1}}} d_\mu(s_1)\left(\prod_{i=1}^n \mu(a_i|s_i)p(s_{i+1},r_{i+1}|s_i,a_i)\right)\left(\frac{1}{n}\sum_{t=1}^n \rho_t\Delta_t\right) \\
&= \frac{1}{n}\sum_{t=1}^n \sum_{\substack{s_1,a_1,r_2,s_2,\ldots,\\ s_n,a_n,r_{n+1},s_{n+1}}} d_\mu(s_1)\left(\prod_{i=1}^n \mu(a_i|s_i)p(s_{i+1},r_{i+1}|s_i,a_i)\right)(\rho_t\Delta_t)\,.
\end{aligned}$$

Next, by taking the sums over $(s_1,a_1,...r_{n+1},s_{n+1})$ within the products to make the summands depend only on the variables being summed over, we get

$$\begin{aligned}
&= \frac{1}{n}\sum_{t=1}^n \sum_{s_1} d_\mu(s_1) \sum_{a_1,r_2,s_2} \mu(a_1|s_1)p(s_2,r_2|s_1,a_1) \sum_{a_2,r_3,s_3} \mu(a_2|s_2)p(s_3,r_3|s_2,a_2)... \\
&\quad \sum_{a_t,r_{t+1},s_{t+1}} \mu(a_t|s_t)p(s_{t+1},r_{t+1}|s_t,a_t)\,(\rho_t\Delta_t) \\
&\quad \sum_{\substack{a_{t+1},r_{t+2},s_{t+2},\ldots,\\ s_n,a_n,r_{n+1},s_{n+1}}} \prod_{i=t+1}^n \mu(a_i|s_i)p(s_{i+1},r_{i+1}|s_i,a_i) \\
&= \frac{1}{n}\sum_{t=1}^n \sum_{s_1} d_\mu(s_1) \sum_{a_1,r_2,s_2} \mu(a_1|s_1)p(s_2,r_2|s_1,a_1) \sum_{a_2,r_3,s_3} \mu(a_2|s_2)p(s_3,r_3|s_2,a_2)... \\
&\quad \sum_{a_t,r_{t+1},s_{t+1}} \mu(a_t|s_t)p(s_{t+1},r_{t+1}|s_t,a_t)\,(\rho_t\Delta_t)\,.
\end{aligned}$$

This followed since the third line is summing over the probability of all trajectories starting from s_{t+1} and thus is equal to 1. Next, we note that, if C is a constant that does not depend on s_1,a_1,r_2, then $\sum_{s_1,a_1,r_2} d_\mu(s_1)\mu(a_1|s_1)p(s_2,r_2|s_1,a_1)C = d_\mu(s_2)C$ since $d_\mu(s_2)$ is the stationary distribution (if we additionally assume $p(s_2,r_2|s_1,a_1) = p(s_2|s_1,a_1)p(r_2|s_1,a_1)$ or equivalently that rewards depend only on state and action).

Continuing from before, by reordering the sums we have and repeatedly using the above note,

$$= \frac{1}{n}\sum_{t=1}^{n}\sum_{s_2}\underbrace{\sum_{s_1,a_1,r_2} d_\mu(s_1)\mu(a_1|s_1)p(s_2,r_2|s_1,a_1)}_{d_\mu(s_2)}\sum_{a_2,r_3,s_3}\mu(a_2|s_2)p(s_3,r_3|s_2,a_2)...$$

$$\sum_{a_t,r_{t+1},s_{t+1}}\mu(a_t|s_t)p(s_{t+1},r_{t+1}|s_t,a_t)\left(\rho_t\Delta_t\right)$$

$$= \frac{1}{n}\sum_{t=1}^{n}\sum_{s_2} d_\mu(s_2)\sum_{a_2,r_3,s_3}\mu(a_2|s_2)p(s_3,r_3|s_2,a_2)...$$

$$\sum_{a_t,r_{t+1},s_{t+1}}\mu(a_t|s_t)p(s_{t+1},r_{t+1}|s_t,a_t)\left(\rho_t\Delta_t\right)$$

$$= ... \quad \text{(Repeating the same process)}$$

$$= \frac{1}{n}\sum_{t=1}^{n}\sum_{s_t,a_t,r_{t+1},s_{t+1}} d_\mu(s_t)\mu(a_t|s_t)p(s_{t+1},r_{t+1}|s_t,a_t)\left(\rho_t\Delta_t\right).$$

Recall that $\Delta_t = \Delta(s_t, a_t, r_{t+1}, s_{t+1})$ is a function of the transition so we cannot simplify further.

Finally,

$$= \frac{1}{n}\sum_{t=1}^{n}\sum_{s_t,a_t,r_{t+1},s_{t+1}} d_\mu(s_t)\mu(a_t|s_t)p(s_{t+1},r_{t+1}|s_t,a_t)\left(\frac{\pi(a_t)}{\mu(a_t)}\Delta_t\right)$$

$$= \frac{1}{n}\sum_{t=1}^{n}\sum_{s_t,a_t,r_{t+1},s_{t+1}} d_\mu(s_t)\pi(a_t|s_t)p(s_{t+1},r_{t+1}|s_t,a_t)\Delta_t$$

$$= \frac{1}{n}\sum_{t=1}^{n}\mathbb{E}_\pi[\Delta]$$

$$= \mathbb{E}_\pi[\Delta]$$

□

Theorem 3.2 Let $B_t = \{X_{t+1}, ..., X_{t+n}\}$ be the buffer of the most recent n transitions sampled by time $t+n$, which are generated sequentially from an irreducible, finite MDP with a fixed policy μ. Define the sliding-window estimator $X_t \overset{\text{def}}{=} \frac{1}{T}\sum_{t=1}^{T} X_{\text{BC}}^{(t)}$. Then, if $\mathbb{E}_\pi[|\Delta|] < \infty$, then X_T converges to $\mathbb{E}_\pi[\Delta]$ almost surely as $T \to \infty$.

Proof. Let $X_t = (S_t, A_t, R_{t+1}, S_{t+1})$ be a transition. Then the sequence $\{X_t\}_{t\in\mathbb{N}}$ forms an irreducible Markov chain as there is positive probability of eventually visiting any X' starting from any X since this is true for states S' and S in the original MDP (by irreducibility).

Let $\{B_t\}_{t\in\mathbb{N}}$ be the sequence of buffers that are observed. This also forms an irreducible Markov chain by the same reasoning as above since $\{X_t\}_{t\in\mathbb{N}}$ is irreducible. Additionally, the sequence of pairs $\{(X_{\text{BC}}^{(t)}, B_t)\}_{t\in\mathbb{N}}$ is an irreducible Markov chain.

Using the ergodic theorem (theorem 4.16 in [Levin and Peres, 2017]) on $\{(X_{\text{BC}}^{(t)}, B_t)\}_{t\in\mathbb{N}}$ with the projection function $f(x, y) = x$, we have that

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T} X_{\text{BC}}^{(t)} = \mathbb{E}\left[X_{\text{BC}}^{(t)}\right]$$

where the expectation is over the joint stationary distribution of $(X_{\text{BC}}^{(t)}, B_t)$.

Using Lemma B.2 we can show that $\mathbb{E}\left[X_{\text{BC}}^{(t)}\right] = \mathbb{E}_\pi[\Delta]$, completing the proof.

□

B.5 Variance of BC-IR and IS

This lemma characterizes the variance of the BC-IR and IS estimators for a fixed buffer.

Lemma B.3. *Let $\mu_B = \mathbb{E}_\pi[\Delta|B]$ be the mean update on the batch B. Denoting the size of the buffer by n and the number of size of the minibatch by k, let $X_{IS} = \frac{1}{k}\sum_{j=1}^{k} \rho_{z_j}\Delta_{z_j}$ (with each z_j sampled uniformly from $\{1, ..., n\}$) be the importance sampling estimator and $X_{BC} = \frac{1}{k}\sum_{j=1}^{n} \Delta_{i_j}$ (with each i_j being sampled from $\ell \in \{1, ..., n\}$ with probability proportional to ρ_ℓ) be the bias-corrected importance resampling estimator. Then, the variances of the two estimators are given by*

$$\mathbb{V}(X_{\mathrm{IS}} \mid B) = \frac{1}{k}\left(\frac{1}{n}\sum_{j=1}^{n} \rho_j^2 \left\|\Delta_j\right\|_2^2 - \mu_B^\top \mu_B\right)$$

$$\mathbb{V}(X_{\mathrm{BC}} \mid B) = \frac{1}{k}\left(\frac{\bar{\rho}}{n}\sum_{j=1}^{n} \rho_j \left\|\Delta_j\right\|_2^2 - \mu_B^\top \mu_B\right)$$

Proof. Since we condition on the buffer B, the only source of randomness is the sampling mechanism. Each index is sampled independently so we have that,

$$\mathbb{V}(X_{\mathrm{BC}} \mid B) = \frac{1}{k^2}\sum_{j=1}^{k} \mathbb{V}(\bar{\rho}\Delta_{i_j} \mid B) = \frac{1}{k}\mathbb{V}(\bar{\rho}\Delta_{i_1} \mid B)$$

and similarly $\mathbb{V}(X_{\mathrm{IS}} \mid B) = \frac{1}{k}\mathbb{V}(\rho_{z_1}\Delta_{z_1}|B)$

We can further simplify these expressions. For the IS estimator

$$\begin{aligned}
&\mathbb{V}(\rho_{z_1}\Delta_{z_1}|B) \\
&= \mathbb{E}[\rho_{z_1}^2 \Delta_{z_1}^\top \Delta_{z_1}|B] - \mathbb{E}[\rho_{z_1}\Delta_{z_1}|B]^\top \mathbb{E}[\rho_{z_1}\Delta_{z_1}|B] \quad \text{by definition of } \mathbb{V}(\cdot) \\
&= \mathbb{E}[\rho_{z_1}^2 \|\Delta_{z_1}\|_2^2|B] - \mu_B^\top \mu_B \quad \text{since } \rho_{z_1}\Delta_{z_1}|B \text{ is unbiased for } \mu_B \\
&= \frac{1}{n}\sum_{j=1}^{n} \rho_j^2 \|\Delta_j\|_2^2 - \mu_B^\top \mu_B
\end{aligned}$$

The last line follows from the uniform sampling distribution. For the BC-IR estimator, recalling that $\bar{\rho} = \frac{1}{n}\sum_{i=1}^{n} \rho_i$, we follow similar steps,

$$\begin{aligned}
&\mathbb{V}(\bar{\rho}\Delta_{i_1}|B) \\
&= \mathbb{E}\left[\bar{\rho}^2 \Delta_{i_1}^\top \Delta_{i_1}|B\right] - \mathbb{E}[\bar{\rho}\Delta_{i_1}|B]^\top \mathbb{E}[\bar{\rho}\Delta_{i_1}|B] \\
&= \mathbb{E}\left[\bar{\rho}^2 \|\Delta_{i_j}\|_2^2|B\right] - \mu_B^\top \mu_B \quad \text{since } \bar{\rho}\Delta_{i_1}|B \text{ is unbiased for } \mu_B \\
&= \sum_{j=1}^{n} \bar{\rho}^2 \frac{\rho_j}{\sum_{i=1}^{n}\rho_i} \|\Delta_j\|_2^2 - \mu_B^\top \mu_B \\
&= \frac{\bar{\rho}}{n}\sum_{j=1}^{n} \rho_j \|\Delta_j\|_2^2 - \mu_B^\top \mu_B
\end{aligned}$$

The fourth line follows from the sampling distribution of the i_j. □

The following two theorems present certain conditions when the BC-IR estimator would have lower variance than the IS estimator.

Theorem 3.3 Assume that $\|\Delta_j\|_2^2 > \frac{c}{\rho_j}$ for samples where $\rho_j \geq \bar{\rho}$, and that $\|\Delta_j\|_2^2 < \frac{c}{\rho_j}$ for samples where $\rho_j < \bar{\rho}$, for some $c > 0$. Then the BC-IR estimator has lower variance than the IS estimator.

Proof. We show $\mathbb{V}(X_{\text{IS}}|B) - \mathbb{V}(X_{\text{BC}}|B) > 0$:

$$\begin{aligned}
\mathbb{V}(X_{\text{IS}}|B) - \mathbb{V}(X_{\text{BC}}|B) &= \frac{1}{nk}\sum_{j=1}^{n} \|\Delta_j\|_2^2 \left(\rho_j^2 - \bar{\rho}\rho_j\right) \\
&= \frac{1}{nk}\sum_{s:\rho_s<\bar{\rho}} \underbrace{\|\Delta_s\|_2^2}_{\leq c/\rho_s} \rho_s \underbrace{\left(\rho_s - \bar{\rho}\right)}_{\leq 0} + \frac{1}{nk}\sum_{l:\rho_l\geq\bar{\rho}} \underbrace{\|\Delta_l\|_2^2}_{>c/\rho_s} \rho_l \underbrace{\left(\rho_l - \bar{\rho}\right)}_{\geq 0} \\
&> \frac{1}{nk}\sum_{s:\rho_s<\bar{\rho}} \frac{c}{\rho_s}\rho_s\left(\rho_s - \bar{\rho}\right) + \frac{1}{nk}\sum_{l:\rho_l\geq\bar{\rho}} \frac{c}{\rho_l}\rho_l\left(\rho_l - \bar{\rho}\right) \\
&= \frac{c}{nk}\sum_{j=1}^{n}(\rho_j - \bar{\rho}) = 0
\end{aligned}$$

□

Theorem 3.4 Assume ρ and the magnitude of the update $\|\Delta\|_2^2$ are independent

$$\mathbb{E}[\rho_j\|\Delta_j\|_2^2 \mid B] = \mathbb{E}[\rho_j \mid B]\,\mathbb{E}[\|\Delta_j\|_2^2 \mid B]$$

Then the BC-IR estimator will have equal or lower variance than the IS estimator.

Proof. Because of the condition, we can further simplify the variance equations from Lemma B.3. Let $c = \mathbb{E}[\|\Delta_j\|_2^2 \mid B]$. Then for BC-IR we have

$$\frac{\bar{\rho}}{nk}\sum_{j=1}^{n}\rho_j\left\|\Delta_j\right\|^2 = \frac{1}{k}\bar{\rho}\mathbb{E}\left[\rho_j\left\|\Delta_j\right\|^2|B\right] = \frac{1}{k}\bar{\rho}\bar{\rho}c = \frac{1}{k}\bar{\rho}^2 c$$

and for IS we have

$$\frac{1}{nk}\sum_{j=1}^{n}\rho_j^2\|\Delta_j\|_2^2 = \frac{1}{k}\mathbb{E}\left[\rho_j^2\left\|\Delta_j\right\|_2^2|B\right] = \frac{c}{k}\mathbb{E}\left[\rho_j^2|B\right]$$

Now when we take the difference, we get

$$\begin{aligned}
\mathbb{V}(X_{\text{IS}}|B) - \mathbb{V}(X_{\text{BC}}|B) &= \frac{c}{k}(\mathbb{E}\left[\rho_j^2|B\right] - \bar{\rho}^2) \\
&= \frac{c}{k}\hat{\sigma}^2(\rho_j \mid B)
\end{aligned}$$

where $\hat{\sigma}^2(\rho_j)$ is the sample variance of the importance weights $\{\rho_j\}_{j=1}^n$ for B. Because the sample variance is greater than zero and $c \geq 0$, the BC-IR estimator will have equal or lower variance than the IS estimator.

□

B.6 Variance of BC-IR and WIS for a fixed dataset

The variance of BC-IR as compared to IS discussed in section 3.3 is only one comparison we can make. Similarly to bias, we can characterize the variance of the IR estimator relative to WIS-Optimal. X_{WIS^*} is able to use a batch update on all the data in the buffer, which should result in a low-variance estimate but is an unrealistic algorithm to use in practice. Instead, it provides a benchmark, where the goal is to obtain similar variance to X_{WIS^*}, but within realistic computational restrictions. Because of the relationship between IR and WIS, as used in Theorem 3.1, we can characterize the variance of X_{IR} relative to X_{WIS^*} using the law of total covariance:

$$\begin{aligned}
\mathbb{V}(X_{\text{IR}}) &= \mathbb{V}\left[\mathbb{E}[X_{\text{IR}}|B]\right] + \mathbb{E}\left[\mathbb{V}[X_{\text{IR}}|B]\right] \\
&= \mathbb{V}\left[X_{\text{WIS}^*}\right] + \mathbb{E}\left[\mathbb{V}[X_{\text{IR}}|B]\right]
\end{aligned}$$

where the variability is due to having randomly sampled buffers B and random sampling from B. The second term corresponds to the noise introduced by sampling a mini-batch of k transitions from the buffer B, instead of using the entire buffer like WIS. For more insight, we can expand

this second term, $\mathbb{E}\left[\mathbb{V}[X_{\text{IR}}|B]\right] = \mathbb{E}\left[(\frac{1}{k}\sum_{j=1}^{k}\Delta_{i_j} - \frac{1}{n}\sum_{i=1}^{n}\Delta_i)^2|B\right]$, where we consider the variance independently for each element of Δ_i and so apply the square element-wise. The variability is not due to IS ratios, and instead arises from variability in the updates themselves. Therefore, the variance of IR corresponds to the variance of WIS, with some additional variance due to this variability around the average update in the buffer.

C Extended Experimental Results

C.1 Markov Chain

This section contains the full set of markov chain experiments using several different policies. Results can be found in figure 4 and figure 6. See figure captions for more details.

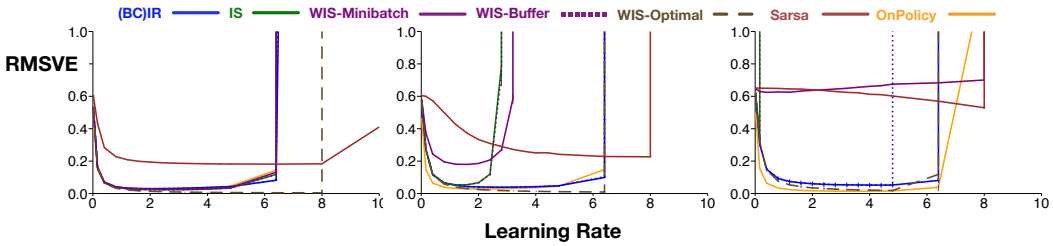

Figure 5: Sensitivity curves for Markov Chain experiments with policy action probabilities [left, right] **left** $\mu = [0.5, 0.5], \pi = [0.1, 0.9]$; **center** $\mu = [0.9, 0.1], \pi = [0.1, 0.9]$; **right** $\mu = [0.99, 0.01], \pi = [0.01, 0.99]$.

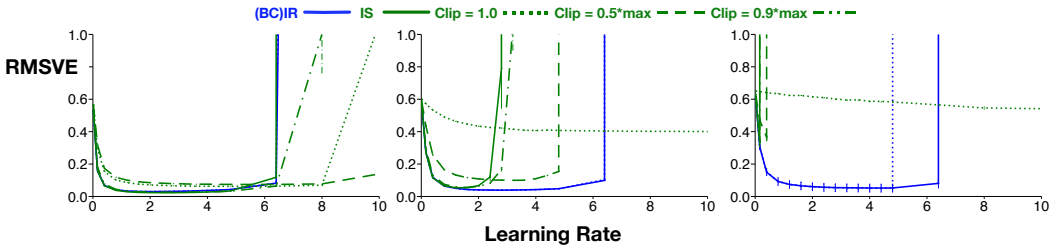

Figure 6: Learning rate sensitivity plots for V-Trace (with the same settings as Figure 4). Three clipping parameters were chosen, including 1.0, 0.5 ρ_{max} and 0.9 ρ_{max}, where ρ_{max} is the maximum possible IS ratio. For 1.0 ρ_{max}, updates under V-trace become exactly equivalent to IS.

C.2 Continuous Four Rooms

The continuous four rooms environment is an 11x11 2d continuous world with walls setup as the original four rooms environment grid world. The agent is a circle with radius 0.1, and the state consists of a continuous tuple containing the x and y coordinates of the agent's center point. The agent takes an action in one of the 4 cardinal directions moving $0.5 \pm \mathbb{U}(0.0, 0.1)$ in that directions and random drift in the orthogonal direction sampled from $\mathbb{N}(0.0, 0.01)$. The simulation takes 10 intermediary steps to more accurately detect collisions.

We use three behavior policies in our experiments. **Uniform:** the agent selects all actions uniformly, **State Variant:** the agent selects all actions uniformly except in pre-determined subsections of the environment where the agent will take down with likelihood 0.1 and the rest distributed evenly over the other actions, **State Weight Variant:** the agent selects all actions uniformly except in pre-determined subsections where the pmf is defined randomly. We also use two target policies. **Persistent Down**: where the agent always takes the down action, **Favored Down:** where the agent takes the down action with likelihood 0.9 and uniformly among the other actions with likelihood 0.1. We use a cumulant function which indicates collision with a wall and a termination function

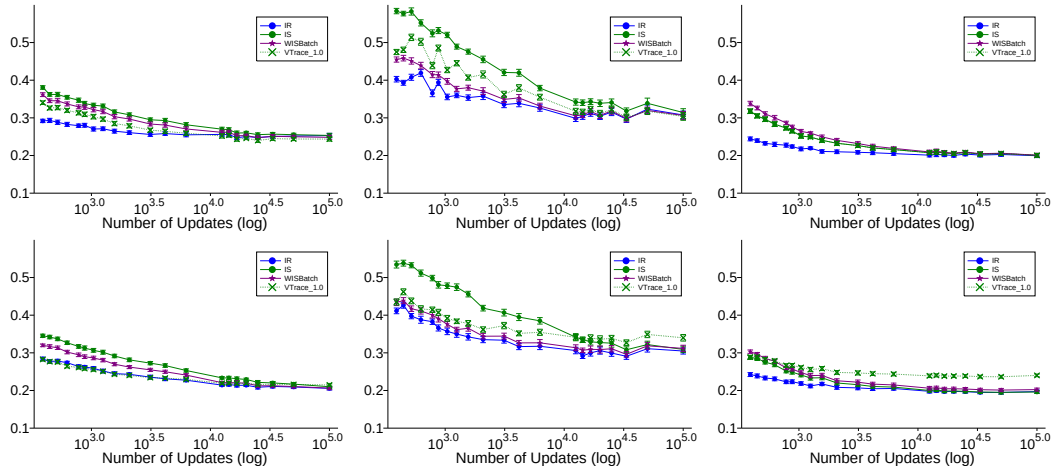

Figure 7: SGD: Target Policy: **Top**: persistent down, **Bottom** favored down. Behaviour Policy: **left** State Variant **center** State Weight Variant **right** Uniform. Sample efficiency plots.

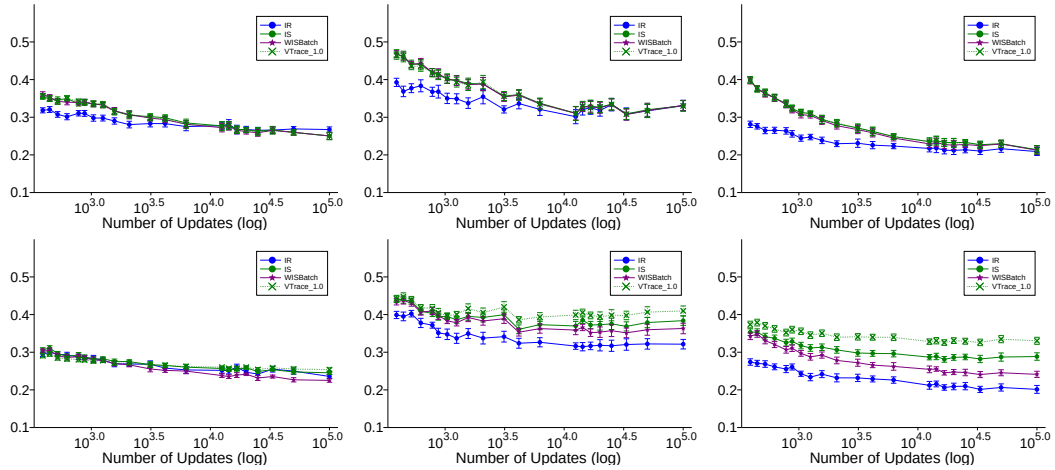

Figure 8: RMSProp Target Policy: **Top**: persistent down, **Bottom** favored down. Behaviour Policy: **left** State Variant **center** State Weight Variant **right** Uniform. Sample efficiency plots.

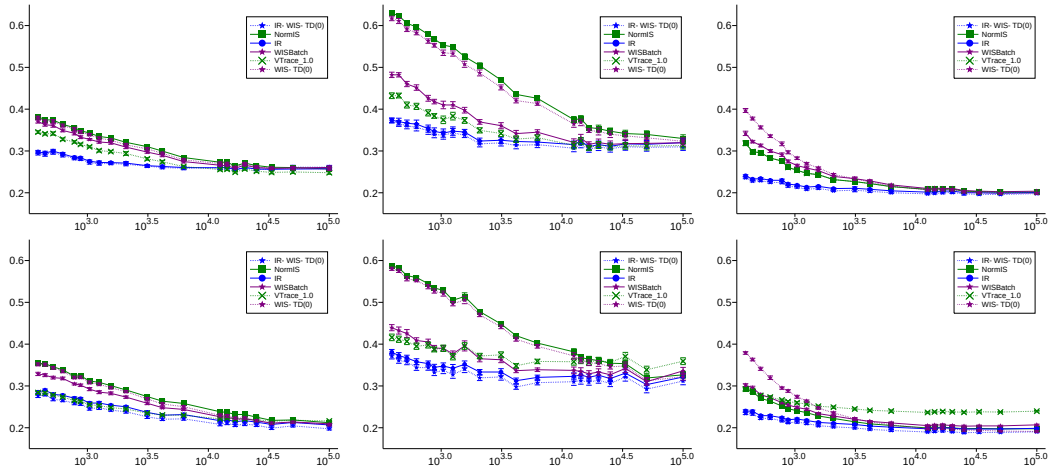

Figure 9: Incremental Experiments Target Policy: **Top**: persistent down, **Bottom** favored down. Behaviour Policy: **left** State Variant **center** State Weight Variant **right** Uniform. Sample efficiency plots.

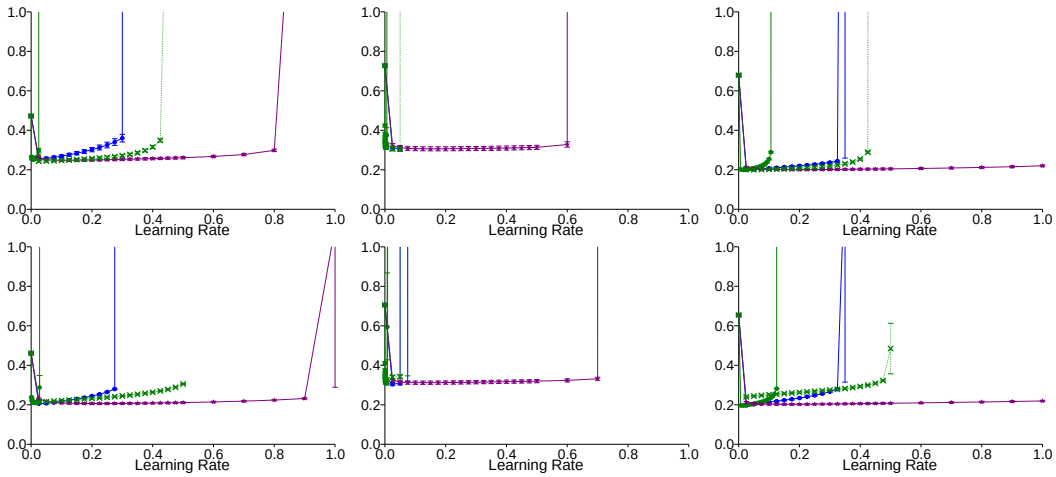

Figure 10: SGD: Target Policy: **Top**: persistent down, **Bottom** favored down. Behaviour Policy: **left** State Variant **center** State Weight Variant **right** Uniform. Learning Rate Sensitivity

which terminates on collision and is 0.9 otherwise for all value functions. We present results using SGD and RMSProp over many algorithms and parameter settings in figures 7, 8, , and 10.