Proceedings of Machine Learning Research vol 168:1–14, 2022 4th Annual Conference on Learning for Dynamics and Control

Experience Replay with Likelihood-free Importance Weights

Samarth Sinha* SAMARTH.SINHA@MAIL.UTORONTO.CA
Jiaming Song* TSONG@CS.STANFORD.EDU
Animesh Garg GARG@CS.TORONTO.EDU
Stefano Ermon ERMON@CS.STANFORD.EDU

Editors: R. Firoozi, N. Mehr, E. Yel, R. Antonova, J. Bohg, M. Schwager, M. Kochenderfer

Abstract

The use of past experiences to accelerate temporal difference (TD) learning of value functions, or experience replay, is a key component in deep reinforcement learning methods such as actor-critic. In this work, we propose to reweight experiences based on their likelihood under the stationary distribution of the current policy, and justify this with a contraction argument over the Bellman evaluation operator. The resulting TD objective encourages small approximation errors on the value function over frequently encountered states. To balance bias (from off-policy experiences) and variance (from on-policy experiences), we use a likelihood-free density ratio estimator between on-policy and off-policy experiences, and use the learned ratios as the prioritization weights. We apply the proposed approach empirically on Soft Actor Critic (SAC), Double DQN and Data-regularized Q (DrQ), over 12 Atari environments and 6 tasks from the DeepMind control suite. We achieve superior sample complexity on 9 out of 12 Atari environments and 16 out of 24 method-task combinations for DCS compared to the best baselines.

Keywords: Experience Replay, Reinforcement Learning, Learning for Control

1. Introduction

Deep reinforcement learning methods have achieved much success in a wide variety of domains (Mnih et al., 2016; Lillicrap et al., 2015; Horgan et al., 2018). While on-policy methods (Schulman et al., 2017) are effective, using off-policy data often yields better sample efficiency (Haarnoja et al., 2018; Fujimoto et al., 2018), which is critical when querying the environment is expensive and experiences are difficult to obtain. Experience replay (Lin, 1992) is a popular paradigm in off-policy reinforcement learning. When applied to temporal difference (TD) learning of the Q-value function (Mnih et al., 2015), the use of replay buffers avoids forgetting of previous experiences and improves learning. Selecting experiences from the replay buffers using a prioritization scheme (instead of uniformly) can lead to large improvements in terms of sample efficiency (Hessel et al., 2017).

Existing prioritization procedures rely on certain choices of importance sampling; for instance, Prioritized Experience Replay (PER) selects experiences with high TD error more often, and then down-weight the experiences that are frequently sampled in order to become closer to uniform sampling over the experiences (Schaul et al., 2015). However, this might not work well in actor-critic methods, where the goal is to learn the value function (or Q-value function) induced by the

. An additional appendix is available here.

current policy, and following off-policy experiences might be harmful. In this case, it might be more beneficial to perform importance sampling that reflects on-policy experiences instead.

Based on this, we investigate a new prioritization strategy based on the likelihood (i.e., the frequency) of experiences under the stationary distribution of the current policy (Tsitsiklis et al., 1997). In actor-critic methods (Konda and Tsitsiklis, 2000), we can estimate the value function of a policy by minimizing the expected squared difference between the critic network and its target value over a replay buffer; an appropriate replay buffer should properly reflect the discrepancy between critic value functions. We treat a discrepancy as "proper" if it preserves the contraction properties of the Bellman operator, and consider discrepancies measured by the expected squared distances under some state-action distribution. In Theorem 2 we prove that the stationary distribution of the current policy is the *only* distribution in which the Bellman operator is a contraction (i.e. being "proper"); this motivates the use of the stationary distribution as the underlying distribution for the replay buffer.

To use replay buffers derived from the stationary distribution with existing deep reinforcement learning methods, we need to be mindful of the following bias-variance trade-off. We have fewer experiences from the current policy (using which results in high variance estimates), but more experiences from other policies under the same environment (using which results in high bias estimates). Inspired by recent advances in inverse reinforcement learning (Fu et al., 2017) and off-policy policy evaluation (Grover et al., 2019), we use a likelihood-free method to obtain an estimate of the density ratio from a classifier trained to distinguish different types of experiences. We consider a smaller, "fast" replay buffer that contains near on-policy experiences, and a larger, "slow" replay buffer that contains additional off-policy experiences, and estimate density ratios between the two buffers. We then use these estimated density ratios as importance weights over the Q-value function update objective. This encourages more updates over state-action pairs that are more likely under the stationary policy distribution of the current policy, i.e., closer to the fast replay buffer.

Our approach can be readily combined with existing approaches that learn value functions from replay buffers. We consider our approach over three competitive actor-critic methods, Soft Actor-Critic (SAC, Haarnoja et al. (2018)), Double DQN Van Hasselt et al. (2016), and Data-regularized Q (DrQ, Kostrikov et al. (2020)). We demonstrate the effectiveness of our approach over on 12 environments from the Atari Arcade Learning Environment (Bellemare et al., 2013) and 6 environments from DeepMind Control Suite (Tassa et al., 2018), where both low-dimensional state space and high-dimensional image space are considered; this results in 36 method-task combinations in total. Notably, our approach outperforms the respective baselines in 25 out of the 36 cases, while being competitive in the remaining 11 cases. This demonstrates that our method can be applied as a simple plug-and-play approach to improve existing actor-critic methods.

2. Preliminaries

The reinforcement learning problem can be described as finding a policy for a Markov decision process (MDP) defined as the following tuple $(\mathcal{S}, \mathcal{A}, P, r, \gamma, p_0)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is the transition kernel, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\gamma \in [0, 1)$ is the discount factor and $p_0 \in \mathcal{P}(\mathcal{S})$ is the initial state distribution. The goal is to learn a stationary policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ that selects actions in $\mathcal{A}$ for each state $s \in \mathcal{S}$, such that the policy maximizes the expected sum of rewards: $J(\pi) := \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$, where the expectation is over trajectories sampled from $s_0 \sim p_0$, $a_t \sim \pi(\cdot|s_t)$, and $s_{t+1} \sim P(\cdot|s_t, a_t)$ for $t \geq 0$.

For a fixed policy, the MDP becomes a Markov chain, so we define the state-action distribution at timestep t: $d_t^\pi(s,a)$, and the the corresponding (unnormalized) stationary distribution over states and actions $d_\pi(s,a) = \sum_{t=0}^{\infty} \gamma^t d_t^\pi(s,a)$ (we assume this always exists for the policies we consider). We can then write $J(\pi) = \mathbb{E}_{d^\pi}[r(s,a)]$. For any stationary policy π, we define its corresponding state-action value function as $Q^\pi(s,a) := \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t)|s_0 = s, a_0 = a]$, its corresponding value function as $V^\pi(s) := \mathbb{E}_{a\sim\pi(\cdot|s)}[Q^\pi(s,a)]$ and the advantage function $A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$.

A large variety of actor-critic methods (Konda and Tsitsiklis, 2000) have been developed in the context of deep reinforcement learning (Silver et al., 2014; Mnih et al., 2016; Lillicrap et al., 2015; Haarnoja et al., 2018; Fujimoto et al., 2018), where learning good approximations to the Q-function is critical to the success of any deep reinforcement learning method based on actor-critic paradigms.

The Q-function can be learned via temporal difference (TD) learning (Sutton, 1988) based on the Bellman equation $Q^\pi(s,a) = \mathcal{B}^\pi Q^\pi(s,a)$; where $\mathcal{B}^\pi$ denotes the Bellman **evaluation** operator

$$\mathcal{B}^\pi Q(s,a) := r(s,a) + \gamma \mathbb{E}_{s',a'}[Q(s',a')], \tag{1}$$

where in the expectation we sample the next step, $s' \sim P(\cdot|s,a)$ and $a' \sim \pi(\cdot|s)$.

Given some experience replay buffer $\mathcal{D}$ (collected by navigating the same environment, but with unknown and potentially different policies), one could optimize the following loss for a Q-network:

$$L_Q(\theta; \mathcal{D}) = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[(Q_\theta(s,a) - \hat{\mathcal{B}}^\pi Q_\theta(s,a))^2\right] \tag{2}$$

which fits $Q_\theta(s,a)$ to an estimate of the target value $\hat{\mathcal{B}}^\pi[Q_\theta](s,a)$[1]. In practice, the target values can be estimated either via on-policy experiences (Sutton et al., 1999) or via off-policy experiences (Precup, 2000). Ideally, we can learn Q^π by optimizing the $L_Q(\theta; \mathcal{D})$ to zero with over-parametrized neural networks. However, instead of minimizing the loss $L_Q(\theta; \mathcal{D})$ directly, prioritization over the sampled replay buffer $\mathcal{D}$ could lead to stronger performance. For example, prioritized experience replay (PER, (Schaul et al., 2015)) is a heuristic that assigns higher weights to transitions with higher TD errors, and is applied successfully in deep Q-learning (Hessel et al., 2017).

In this paper, we discuss actor-critic methods, whose target is fundamentally different from that of Q-learning (which aims to learn the "optimal" Q-function):

$$\mathcal{B}^\star = r(s,a) + \gamma \max_{a'\in\mathcal{A}}[Q(s',a')]. \tag{3}$$

where $\mathcal{B}^\star$ denotes the Bellman **optimality** operator. As we will show in the experiments, prioritization strategies developed specifically for Q-learning are not well suited for policy gradient / actor-critic methods we consider here.

3. Experience Replay based on Stationary Distributions

Assume that d, the distribution the replay buffer $\mathcal{D}$ is sampled from, is supported on the entire space $\mathcal{S} \times \mathcal{A}$, and that we have infinite samples from π (so the Bellman target is unbiased). Let us define the TD-learning objective for Q with prioritization weights $w : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^+$, under the sampling distribution $d \in \mathcal{P}(\mathcal{S} \times \mathcal{A})$:

$$L_Q(\theta; d, w) = \mathbb{E}_d\left[w(s,a)(Q_\theta(s,a) - \mathcal{B}^\pi Q_\theta(s,a))^2\right] \tag{4}$$

1. We also do not take the gradient over the target, which is the more conventional approach.

In practice, the expectation in $L_Q(\theta; d, w)$ can be estimated with Monte-Carlo methods, such as importance sampling, rejection sampling, or combinations of multiple methods (such as in PER (Schaul et al., 2015)). Without loss of generality, we can treat the problem as optimizing the mean squared TD error under some *priority distribution* $d^w \propto d \cdot w$, so one could treat prioritized experience replay for TD learning as selecting a favorable *priority distribution* d^w (under which the L_Q loss is computed) in order to improve some notion of performance.

In this paper, we propose to use as *priority distribution* $d^w = d^\pi$, where d^π is the stationary distribution of state-action pairs under the current policy π. This reflects the intuition that TD-errors in high-frequency state-action pairs are more problematic than in low-frequency ones, as they will negatively impact policy updates more severely. In the following subsection, we argue the importance of choosing d^π from the perspective of maintaining desirable contraction properties of the Bellman operators under more general norms. If we consider Euclidean norms weighted under some distribution $d^w \in \mathcal{P}(\mathcal{S} \times \mathcal{A})$, the usual γ-contraction argument for Bellman operators holds only for $d^w = d^\pi$, and not for other distributions.

Policy-dependent Norms for Bellman Backup The convergence of Bellman updates relies on the fact that the Bellman evaluation operator $\mathcal{B}^\pi$ is a γ-contraction with respect to the ℓ_∞ norm, i.e. $\forall Q, Q' \in \mathcal{Q}$, where $\mathcal{Q} = \{Q : (\mathcal{S} \times \mathcal{A}) \to \mathbb{R}\}$ is the set of all possible Q functions:

$$\|\mathcal{B}^\pi Q - \mathcal{B}^\pi Q'\|_\infty \leq \gamma \|Q - Q'\|_\infty \tag{5}$$

While it is sufficient to show convergence results, the ℓ_∞ norm reflects a distance over two Q functions under the worst possible state-action pair, and is independent of the current policy. If two Q functions are equal everywhere except for a large difference on a single state-action pair $(\widetilde{s}, \widetilde{a})$ that is unlikely under d^π, the ℓ_∞ distance between the two Q functions is large. In practice, however, this will have little effect over policy updates as it is unlikely for the current policy to sample $(\widetilde{s}, \widetilde{a})$.

Since our goal with the TD updates is to learn Q^π, a distance metric that is related to π is a more suitable one for comparing different Q functions, reflecting the intuition that errors in frequent state-action pairs are more costly than on infrequent ones. Let us consider the following weighted ℓ_2 distance between Q functions,

$$\|Q - Q'\|_d^2 := \mathbb{E}_{(s,a)\sim d}[(Q(s,a) - Q'(s,a))^2] \tag{6}$$

where $d \in \mathcal{P}(\mathcal{S} \times \mathcal{A})$ is a distribution over state-action pairs. This can be treated as the ℓ_2 norm but measured over stationary distribution d as opposed to the Lebesgue measure. This is closely tied to the L_Q objective since $L_Q(\theta; d) = \|Q_\theta(s,a) - \mathcal{B}^\pi Q_\theta(s,a)\|_d^2$. In the following statements, we show that $\mathcal{B}^\pi$ is only a contraction operator when under the $\|\cdot\|_{d^\pi}$ norm; this supports the use of d^π instead of other distributions for the L_Q objective, as it reflects a more reasonable measurement of distance between Q-functions for policy π.

Lemma 1 *For all $\gamma \in (0,1)$, the Bellman operator $\mathcal{B}^\pi$ is a γ-contraction with respect to the $\|\cdot\|_d$ norm **if** $d = d^\pi$ holds almost everywhere, i.e., $\forall Q, Q' \in \mathcal{Q}$*

$$d = d^\pi \ a.e. \implies \|\mathcal{B}^\pi Q - \mathcal{B}^\pi Q'\|_d \leq \gamma \|Q - Q'\|_d$$

Proof In Appendix A. On a high-level, we apply Jensen's inequality to $f(x) = x^2$. ■

Theorem 2 *For all $\gamma \in (0, 1)$, the Bellman operator $\mathcal{B}^\pi$ is a γ-contraction with respect to the $\|\cdot\|_d$ norm **if and only if** $d = d^\pi$ holds almost everywhere, i.e., $\forall Q, Q' \in \mathcal{Q}$*

$$d = d^\pi, \; a.e. \iff \|\mathcal{B}^\pi Q - \mathcal{B}^\pi Q'\|_d \leq \gamma \|Q - Q'\|_d$$

Proof In Appendix A. On a high-level, whenever $d = d^\pi$ does not hold over some non-empty open set, we can perturb a constant Q-value function over this set to contradict γ-contraction. ∎

Theorem 2 highlights the importance of using d^π in the $\|\cdot\|_d$ norm specifically for measuring the distance between Q-functions: if we use any distribution other than d^π, the Bellman operator is not guaranteed to be a γ-contraction under that distance, which leads to worse convergence rates.

4. Likelihood-free Importance Weighting over Replay Buffers

In practice, however, there are two challenges with regards to using $L_Q(\theta; d^\pi)$ as the objective. On the one hand, an accurate estimate of d^π requires many on-policy samples from d^π and interactions with the environment, which could increase the practical sample complexity; on the other hand, if we instead use off-policy experiences from the replay buffer, it would be difficult to estimate the importance ratio $w(s, a) := d^\pi(s, a)/d^D(s, a)$ when the replay buffer $\mathcal{D}$ is a mixture of trajectories from different policies.

An appropriate choice of importance weights should us to balance **bias** (which comes from replay experiences of other policies) and **variance** (which comes from a small number of on-policy experiences). Thus, we consider likelihood-free density ratio estimation methods that rely only on samples (e.g. from the replay buffer), which are well-suited for estimating the objective function $L_Q(\theta; d^\pi)$ with a good bias-variance trade-off.

4.1. Likelihood-free importance weights

For any convex, lower-semicontinuous function $f : [0, \infty) \to \mathbb{R}$ satisfying $f(1) = 0$, the f-divergence between two probabilistic measures $P, Q \in \mathcal{P}(\mathcal{X})$ (where we assume $P \ll Q$, i.e. P is absolutely continuous w.r.t. Q) is defined as: $D_f(P\|Q) = \int_\mathcal{X} f(\mathrm{d}P(x)/\mathrm{d}Q(x))\, \mathrm{d}Q(x)$. A general variational method can be used to estimate f-divergences given only samples from P and Q.

Lemma 3 (Nguyen et al. (2008)) *Assume that f has first order derivatives f' at $[0, +\infty)$. $\forall P, Q \in \mathcal{P}(\mathcal{X})$ such that $P \ll Q$ and $w : \mathcal{X} \to \mathbb{R}^+$, $D_f(P\|Q) \geq \mathbb{E}_P[f'(w(x))] - \mathbb{E}_Q[f^*(f'(w(x)))]$, where f^* denotes the convex conjugate and the equality is achieved when $w = \mathrm{d}P/\mathrm{d}Q$.*

The above lemma suggests that we can estimate importance weights from samples by optimizing a lower bound to f-divergence. This has been applied to off-policy policy evaluation (Grover et al., 2019), but not directly to actor-critic methods.

4.2. Importance weights for actor-critic methods

We can apply this approach to estimating the likelihood ratio $w(s, a) := d^\pi(s, a)/d^D(s, a)$ with samples from the replay buffer, for both **continuous and discrete** spaces. These ratios are then multiplied to the Q-function updates to perform importance weighting.

To implement this idea in practice, we consider sampling from two types of replay buffers. One is the *regular (slow) replay buffer*, which contains a mixture of trajectories from different policies;

the other is a *smaller (fast) replay buffer*, which contains only a small set of trajectories from very recent policies. After each episode of environment interaction, we update both replay buffers with the new experiences; the distribution of the slow replay buffer changes more slowly due to the larger size. The slow replay buffer contains off-policy samples from d^D whereas the fast replay buffer contains (approximately) on-policy samples from d^π (assuming the buffer size is small enough). Therefore, the slow replay buffer has better coverage of transition dynamics of the environment while being less on-policy. Denoting the fast and slow replay buffers as $\mathcal{D}_\mathrm{f}$ and $\mathcal{D}_\mathrm{s}$ respectively, we estimate the ratio d^π/d^D via minimizing the following objective over the network $w_\psi(x)$ parametrized by ψ (the outputs $w_\psi(s,a)$ are forced to be non-negative via activation functions):

$$L_w(\psi) := \mathbb{E}_{\mathcal{D}_\mathrm{s}}[f^*(f'(w_\psi(s,a)))] - \mathbb{E}_{\mathcal{D}_\mathrm{f}}[f'(w_\psi(s,a))] \tag{7}$$

From Lemma 3, we can recover an estimate of the density ratio from the optimal w_ψ by minimizing the $L_w(\psi)$ objective. To address the finite sample size issue, we apply self-normalization (Cochran, 2007) to the importance weights over the slow replay buffer $\mathcal{D}_\mathrm{s}$ with a hyperparameter T:

$$\tilde{w}_\psi(s,a) := w_\psi(s,a)^{1/T} / \mathbb{E}_{\mathcal{D}_\mathrm{s}}[w_\psi(s,a)^{1/T}] \tag{8}$$

We note that this density ratio is not unbounded, as the slow replay buffer contains all the examples from the fast replay buffer. The final objective for learning Q is then

$$L_Q(\theta; d^\pi) \approx L_Q(\theta; \mathcal{D}_\mathrm{s}, \tilde{w}_\psi) := \mathbb{E}_{(s,a)\sim\mathcal{D}_\mathrm{s}}[\tilde{w}_\psi(x)(Q_\theta(s,a) - \hat{\mathcal{B}}^\pi Q_\theta(s,a))^2]$$

where the target $\hat{\mathcal{B}}^\pi Q_\theta$ is estimated from past experiences. We keep the remainder of the algorithm, such as policy gradient and value network update (if available) unchanged, so this method can be adapted for different off-policy actor-critic algorithms, utilizing their respective advantages. We describe a general procedure of our approach in Algorithm **??**, where one can modify from some "base" actor-critic algorithm to implement our approach. These algorithm cover both stochastic and deterministic policies, as our method does not require likelihood estimates from the policy. We consider our divergences to be Jensen-Shannon, so w_ψ can be treated as a probabilistic classifier.

5. Related Work

Experience replay (Lin, 1992) is a crucial component in deep reinforcement learning (Hessel et al., 2017; Andrychowicz et al., 2017; Schaul et al., 2015), where off-policy experiences are utilized to improve sample efficiency. These experiences can be utilized on policy updates (such as in actor-critic methods (Konda and Tsitsiklis, 2000; Wang et al., 2016)), on value updates (such as in deep Q-learning (Schaul et al., 2015)) or on evaluating TD update targets (Precup, 2000; Precup et al., 2001). For value updates, there are two sources of randomness that could benefit from importance weights (prioritization). The first source is the evaluation of the TD learning target for longer traces such as TD(λ); importance weights can be used to debias targets computed from off-policy trajectories (Precup, 2000; Munos et al., 2016; Espeholt et al., 2018; Schmitt et al., 2019), similar to its role in policy learning. The second source is the sampling of state-action pairs where the values are updated (Schaul et al., 2015), which is addressed in this paper.

Numerous techniques have achieved superior sample complexity through prioritization of replay buffers. In model-based planning, Prioritized Sweeping (Moore and Atkeson, 1993; Andre et al., 1998; van Seijen and Sutton, 2013) selects the next state updates according to changes in value.

Table 1: Results of SAC and TD3 trained from states on the DeepMind Control environments with and without LFIW after 100k and 250k environment steps. **The results show significant improvements when the agents is trained with LFIW.** Results are reported over 5 random seeds. The maximum possible score for any environment is 1,000.

100k environment steps					
SAC based	**SAC**	**DisCor**	**+PER**	**+PER+LFIW**	**+LFIW**
Finger, Spin	482 ± 34	389 ± 29	486 ± 18	503 ± 27	**523** ± 16
Cartpole, Swing	700 ± 51	681 ± 35	689 ± 39	726 ± 14	**789** ± 27
Reacher, Easy	750 ± 68	833 ± 17	704 ± 89	806 ± 55	**861** ± 29
Cheetah, Run	498 ± 108	518 ± 90	367 ± 123	502 ± 109	**541** ± 89
Walker, Walk	187 ± 89	156 ± 57	234 ± 31	**321** ± 29	**333** ± 12
Ball in Cup, Catch	**888** ± 13	876 ± 11	834 ± 23	**892** ± 8	**890** ± 6
250k environment steps					
SAC based	**SAC**	**DisCor**	**+PER**	**+PER+LFIW**	**+LFIW**
Finger, Spin	806 ± 47	800 ± 23	814 ± 45	860 ± 23	**901** ± 14
Cartpole, Swing	825 ± 8	834 ± 21	811 ± 15	823 ± 31	**873** ± 23
Reacher, Easy	**945** ± 32	**940** ± 18	**931** ± 11	**944** ± 6	**941** ± 21
Cheetah, Run	638 ± 32	618 ± 41	675 ± 34	631 ± 56	**709** ± 11
Walker, Walk	895 ± 47	881 ± 23	901 ± 10	**917** ± 20	**911** ± 12
Ball in Cup, Catch	**974** ± 13	**976** ± 7	**978** ± 7	**970** ± 7	**981** ± 19

Table 2: Results for DrQ (Kostrikov et al., 2020) on the image-based RL on the DeepMind Control Suite. LFIW is applied to a state-of-the-art image-based RL algorithm in DrQ, and we are able to see consistent improvement over the DM Control Suite Benchmark.

100k steps	**DrQ**	**DrQ+LFIW**	**500k steps**	**DrQ**	**DrQ+LFIW**
Finger, Spin	838 ± 58	**909** ± 28	Finger, Spin	**918** ± 49	**922** ± 28
Cartpole, Swing	748 ± 50	**801** ± 22	Cartpole, Swing	875 ± 6	**893** ± 8
Reacher, Easy	573 ± 67	**743** ± 89	Reacher, Easy	**945** ± 25	**939** ± 12
Cheetah, Run	387 ± 45	**444** ± 38	Cheetah, Run	**574** ± 104	**581** ± 112
Walker, Walk	639 ± 99	**718** ± 86	Walker, Walk	**901** ± 35	**909** ± 38
Ball in Cup, Catch	**901** ± 17	**901** ± 25	Ball in Cup, Catch	**970** ± 4	**968** ± 8

Prioritized Experience Replay (PER, (Schaul et al., 2015)) emphasizes experiences with larger TD errors and is critical to the success of sample efficient deep Q-learning (Hessel et al., 2017). Remember and Forget Experience Replay (ReF-ER, (Novati and Koumoutsakos, 2018)) removes the experiences if it differs much from choices of the current policy; this encourages sampling on-policy behavior which is similar to what we propose. Differing from ReF-ER, we do not require knowledge of the policy distribution. Distribution Correction (DisCor, Kumar et al. (2020)) suggests against using on-policy experiences, which seems to be in contrast to what we have promoted. However, their analysis is based on the Bellman *optimality* operator, which aims to find the optimal Q-value function, while ours is based on the Bellman *evaluation* operator, which aims to find the Q-value function under the current policy; this could partially explain why DisCor did not achieve superior performance than the baseline approach on OpenAI gym tasks.

Likelihood-free density ratio estimation have been adopted in imitation learning Ho and Ermon (2016), inverse reinforcement learning (Fu et al., 2017), meta learning Fakoor et al. (2019) and model-based off-policy policy evaluation (Grover et al., 2019). Different from these cases, we do not use the weights to estimate the advantage function or to reduce bias in reward estimation; our goal is to improve performance of TD learning with function approximation. Dual representations of f-divergences are also leveraged in reinforcement learning (Nachum et al., 2019; Nachum and Dai, 2020), but it is used over a regularizer that encourages exploration to be closer to off-policy experiences; the importance weights are added to the reward function when computing the Q-value function but do not affect the replay experiences otherwise.

6. Experiments

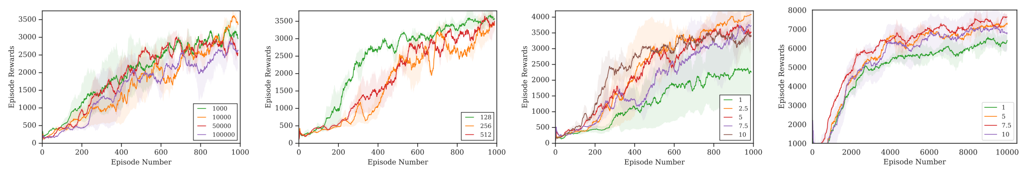
We combine the proposed prioritization approach over three popular actor-critic algorithms, namely Soft-Actor Critic (SAC, Haarnoja et al. (2018)), and Data-regularized Q (DrQ, Kostrikov et al. (2020)); we also applied our method to Double DQN Heess et al. (2015). We compare our method with alternative approaches to prioritization; these include uniform sampling over the replay buffer and prioritization experience replay based on TD-error (Schaul et al., 2015). We choose 12 environments from the Arcade Learning Environment (Atari, (Bellemare et al., 2013)) and 6 tasks from the DeepMind Control suite (DCS, Tassa et al. (2018)). We consider state representations in all tasks and pixel representations from DCS.

Our method introduces some additional hyperparameters compared to the vanilla approaches, namely the temperature T, the size of the fast replay buffer $|\mathcal{D}_{\mathrm{f}}|$ and the architecture for the density estimator w_ψ. To ensure fair comparisons against the baselines, we use the same hyperparameters as the original algorithms when it is available. For all environments we use the following default hyperparameters for likelihood-free importance weighting: $T = 5$, $|\mathcal{D}_{\mathrm{f}}| = 10^4$, $|\mathcal{D}_{\mathrm{s}}| = 10^6$. We use f from the Jensen Shannon divergence for better numerical stability. We include more experimental details in Appendix C.

6.1. Evaluation

Table 3: Results on OpenAI Gym when trained with 500k steps. ERE is only designed for SAC, so its results on TD3 are not available.

SAC based	**SAC**	**+DisCor**	**+PER**	**+ERE**	**+LFIW**
Ant-v2	3193 ± 404	3211 ± 271	2764 ± 287	3331 ± 298	$\mathbf{3579} \pm 260$
HalfCheetah-v2	8325 ± 408	8147 ± 322	8111 ± 341	8631 ± 189	$\mathbf{9045} \pm 222$
Hopper-v2	2645 ± 310	2790 ± 273	2871 ± 214	2512 ± 301	$\mathbf{3109} \pm 244$
Humanoid-v2	2033 ± 199	2569 ± 206	1459 ± 208	2466 ± 147	$\mathbf{3189} \pm 231$
Walker2d-v2	2914 ± 189	2764 ± 166	3071 ± 109	2990 ± 217	$\mathbf{3221} \pm 149$
TD3 based	**TD3**	**+DisCor**	**+PER**	**+ERE**	**+LFIW**
Ant-v2	2663 ± 372	N/A	2610 ± 128	N/A	$\mathbf{2990} \pm 178$
HalfCheetah-v2	7527 ± 438		7310 ± 339		$\mathbf{8567} \pm 491$
Hopper-v2	1801 ± 206		$\mathbf{2019} \pm 109$		1937 ± 250
Walker2d-v2	1306 ± 257		1241 ± 122		$\mathbf{2113} \pm 310$



(a) $|\mathcal{D}_\mathrm{f}|$ (b) w_ψ (c) Temperature T (d) T (Humanoid-v2)

Figure 1: Hyperparameter sensitivity analyses on Walker2d-v2 (a, b, c) and Humanoid-v2 (d) using a base SAC agent and LFIW.

We use (+LFIW) to denote our likelihood-free importance weighting method, (+PER) to denote prioritization with TD error (Schaul et al., 2015)[2] and (+ERE) to denote Emphasizing Recent Experience (ERE, Wang and Ross (2019)) for SAC only. Table 3 shows the results on OpenAI gym (500k steps), whereas Tables 1 and 2 shows the results on DMCS with state (100k and 250k steps) and image representations (100k and 500k steps) respectively. These steps are chosen to demonstrate both initial training progress and approximate performance at convergence.

Table 4: Normalized scores on 12 Atari environments. The baseline numbers are taken directly from Prioritized Experience Replay (Schaul et al., 2015).

	Double DQN			
Environment		+ PER (Rank-based)	+ PER (Proportional)	+ LFIW
Assault	1276%	1381%	1641%	**1889%**
Beam Rider	117%	210%	176%	**217%**
Breakout	1397%	1298%	1407%	**1578%**
Enduro	158%	233%	**239%**	220%
Gopher	728%	1679%	**2792%**	2561%
Ice Hockey	71%	93%	85%	**97%**
Phoenix	202%	284%	474%	**513%**
Pong	**111%**	110%	110%	106%
Q Bert	91%	82%	93%	**103%**
Robotank	872%	828%	815%	**961%**
Video Pinball	7221%	5721%	7367%	**8115%**
Wizard of Wor	144%	131%	177%	**202%**

Table 1 and 2 shows the results with SAC on state representations and DrQ on pixel representations. Again, we observe improvements over the baselines in most cases, and comparable performance in others. Notably, we achieve much higher performance with LFIW at 100k training steps, which demonstrates that biasing the replay buffer towards on-policy experiences is able to achieve good policy performance more quickly. Table 4 shows the results on Atari, where the base algorithm is Double DQN which was trained for 200M time steps using the same hyperparameters as Schaul et al. (2015). Again, we observe that our LFIW-based method outperforms both types of PER on most environments, as well as the base Double DQN agent.

2. We use $\alpha = 0.6, \beta = 0.4$ in PER.

6.2. Additional analyses

Accuracy of w_ψ We use w_ψ to discriminate two types of experiences; experiences sampled from the policy trained with SAC for 5M steps are labeled positive, and the mixture of experiences sampled from policies trained for 1M to 4M steps are labeled negative. With the w_ψ predictions, we obtain a precision of 87.3% and an accuracy of 73.1%. This suggests that the importance weights tends to be higher for on-policy data as desired, thereby making the replay buffer to be closer to on-policy data.

Quality of Q-estimates We compare the quality of the Q-estimates between SAC and SAC+LFIW, where we sample 20 trajectories from each policy, and obtain the "ground truth" via Monte Carlo estimates of the true Q-value. We then evaluate the learned Q-function estimates and compare their correlations with the ground truth values. For the SAC case, the Pearson and Spearman correlations are 0.41 and 0.11 respectively, whereas for SAC+LFIW method they are 0.74 and 0.42 (higher is better). This shows how our Q-function estimates are much more reflective of the "true" values, which explains the improvements in sample complexity and the performance of the learned policy.

6.3. Ablation studies

To study the stability of LFIW across hyperparameters, we conduct further analyses by varying: temperature T in Eq. 8, size of the fast replay buffer $|\mathcal{D}_\mathrm{f}|$, and the number of hidden units in w_ψ. We run SAC+LFIW on Walker-v2 using default hyperparameters, unless stated otherwise.

Temperature T: The temperature T affects the variances of the weights assigned. Since we are using finite replay buffers, using a larger temperature reduces the chances of negatively impacting performance due to w_ψ overfitting the data. We consider $T = 1, 2.5, 5, 7.5, 10$ in Figure 1(c); all cases have similar sample efficiencies except for $T = 1$. Similarly, we also perform a similar analysis on Humanoid-v2 with SAC in Figure 1(d). We observe a similar dependency on T as in Walker where the sample efficiency with $T = 1$ is significantly worse that for the other hyperparameters considered, which shows that overfitting the data can easily be avoided by using a higher temperature value even in higher-dimensional state-action distributions.

Replay buffer sizes $|\mathcal{D}_\mathrm{f}|$: The replay buffer sizes $|\mathcal{D}_\mathrm{f}|$ affects the amount of experiences we treat as "on-policy". Larger $|\mathcal{D}_\mathrm{f}|$ reduces the risk of overfitting while increasing the chances of including more off-policy data. We consider $|\mathcal{D}_\mathrm{f}| = 1000, 10000, 50000, 100000$, corresponding to 1 to 100 episodes. We note that $|\mathcal{D}_\mathrm{s}| = 10^6$, so even for the largest $\mathcal{D}_\mathrm{f}$, $\mathcal{D}_\mathrm{s}$ is significantly larger. The performance are relatively stable despite a small drop for $|\mathcal{D}_\mathrm{f}| = 100000$.

Hidden units of w_ψ: The number of hidden units affects the expressiveness of the neural network as networks with more hidden units are more likely to overfit to the replay buffers. We consider hidden layers with $128, 256$ and 512 neurons respectively. While the smaller network with 128 units is able to achieve superior performance initially, while others catch up at around 1000 episodes.

7. Conclusion

In this paper, we propose a principled approach to prioritized experience replay for actor-critic methods. To achieve a good bias-variance trade-off, we assign weights to the replay buffer based on their estimated density ratios against the stationary distribution. These density ratios are estimated via samples from fast and slow replay buffers, which reflect on-policy and off-policy experiences respectively. Our methods can be readily applied to deep reinforcement learning methods based on actor-critic approaches, such as SAC and DrQ.

Acknowledgments

This research was supported by NSF (#1651565, #1522054, #1733686), ONR (N00014-19-1-2145), AFOSR (FA9550-19-1-0024), ARO (W911NF-21-1-0125), Sloan Fellowship, and Stanford Institute for Human-Centered Artificial Intelligence (HAI). The authors thank Nvidia for donating DGX-1, and Vector Institute for providing resources for this research. Animesh Garg is supported by CIFAR AI Chair, NSERC Discovery Award, University of Toronto XSeed award, and grants from LG and Huawei.

References

David Andre, Nir Friedman, and Ronald Parr. Generalized prioritized sweeping. In M I Jordan, M J Kearns, and S A Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 1001–1007. MIT Press, 1998.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in neural information processing systems*, pages 5048–5058, 2017.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279, 2013.

William G Cochran. *Sampling techniques*. John Wiley & Sons, 2007.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Volodymir Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: Scalable distributed Deep-RL with importance weighted Actor-Learner architectures. *arXiv preprint arXiv:1802.01561*, February 2018.

Rasool Fakoor, Pratik Chaudhari, Stefano Soatto, and Alexander J Smola. Meta-q-learning. *arXiv preprint arXiv:1910.00125*, 2019.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, October 2017.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in Actor-Critic methods. *arXiv preprint arXiv:1802.09477*, February 2018.

Aditya Grover, Jiaming Song, Alekh Agarwal, Kenneth Tran, Ashish Kapoor, Eric Horvitz, and Stefano Ermon. Bias correction of learned generative models using Likelihood-Free importance weighting. *arXiv preprint arXiv:1906.09531*, June 2019.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-Policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, January 2018.

Nicolas Heess, Gregory Wayne, David Silver, Tim Lillicrap, Tom Erez, and Yuval Tassa. Learning continuous control policies by stochastic value gradients. In C Cortes, N D Lawrence, D D Lee, M Sugiyama, and R Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2944–2952. Curran Associates, Inc., 2015.

Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. *arXiv preprint arXiv:1710.02298*, October 2017.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pages 4565–4573, 2016.

Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado van Hasselt, and David Silver. Distributed prioritized experience replay. *arXiv preprint arXiv:1803.00933*, March 2018.

Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

Aviral Kumar, Abhishek Gupta, and Sergey Levine. Discor: Corrective feedback in reinforcement learning via distribution correction. *arXiv preprint arXiv:2003.07305*, 2020.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509. 02971*, 2015.

Long-Ji Lin. Self-Improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8(3):293–321, May 1992. ISSN 0885-6125, 1573-0565. doi: 10.1023/A:1022628806385.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937, 2016.

Andrew W Moore and Christopher G Atkeson. Prioritized sweeping: Reinforcement learning with less data and less time. *Machine learning*, 13(1):103–130, 1993.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc G Bellemare. Safe and efficient Off-Policy reinforcement learning. *arXiv preprint arXiv:1606.02647*, June 2016.

Ofir Nachum and Bo Dai. Reinforcement learning via Fenchel-Rockafellar duality. *arXiv preprint arXiv:2001.01866*, January 2020.

Ofir Nachum, Bo Dai, Ilya Kostrikov, Yinlam Chow, Lihong Li, and Dale Schuurmans. AlgaeDICE: Policy gradient from arbitrary experience. *arXiv preprint arXiv:1912.02074*, December 2019.

Xuanlong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *arXiv preprint arXiv:0809.0853*, (11): 5847–5861, September 2008. doi: 10.1109/TIT.2010.2068870.

Guido Novati and Petros Koumoutsakos. Remember and forget for experience replay. *arXiv preprint arXiv:1807.05827*, July 2018.

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

Doina Precup, Richard S Sutton, and Sanjoy Dasgupta. Off-policy temporal-difference learning with function approximation. In *ICML*, pages 417–424, 2001.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, November 2015.

Simon Schmitt, Matteo Hessel, and Karen Simonyan. Off-Policy Actor-Critic with shared experience replay. *arXiv preprint arXiv:1909.11583*, September 2019.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, July 2017.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International Conference on Machine Learning*, pages 387–395, January 2014.

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

Richard S Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999. ISSN 0004-3702.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

John N Tsitsiklis, Member, and Benjamin Van Roy. An analysis of Temporal-Difference learning with function approximation. *IEEE transactions on automatic control*, 42(5), 1997. ISSN 0018-9286.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Harm van Seijen and Richard S Sutton. Planning by prioritized sweeping with small backups. *arXiv preprint arXiv:1301.2343*, January 2013.

Che Wang and Keith Ross. Boosting soft Actor-Critic: Emphasizing recent experience without forgetting the past. *arXiv preprint arXiv:1906.04009*, June 2019.

Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. Sample efficient Actor-Critic with experience replay. *arXiv preprint arXiv:1611.01224*, November 2016.