Actor Prioritized Experience Replay

Anonymous Author(s)
Affiliation
Address
email

Abstract

A widely-studied deep reinforcement learning (RL) technique known as Prioritized Experience Replay (PER) allows agents to learn from transitions sampled with non-uniform probability proportional to their temporal-difference (TD) error. Although it has been shown that PER is one of the most crucial components for the overall performance of deep RL methods in discrete action domains, many empirical studies indicate that it considerably underperforms actor-critic algorithms in continuous control. We theoretically show that actor networks cannot be effectively trained with transitions that have large TD errors. As a result, the approximate policy gradient computed under the Q-network diverges from the actual gradient computed under the optimal Q-function. Motivated by this, we introduce a new branch of improvements to PER for actor-critic methods, which also regards issues with stability and recent findings behind the poor empirical performance of the algorithm. An extensive set of experiments verifies our theoretical claims and demonstrates that the introduced method obtains substantial gains over PER.

1 Introduction

The use of priority-based non-uniform sampling in deep reinforcement learning (RL) stems from a technique known as Prioritized Experience Replay (PER) [30], in which high error transitions are sampled with higher likelihood, allowing for faster learning and reward propagation by focusing on the most crucial data. In an ablation study, PER was shown to be the most key enhancement for the overall performance of the Deep Q-Network (DQN) algorithm [24] compared to other improvements [16]. Although the motivation of PER is intuitive for learning in discrete action spaces, many empirical studies showed that it notably decreases the performance in continuous action domains, resulting in suboptimal or random behavior [11, 27, 28]. Unfortunately, the poor performance of PER in continuous domains lacks a critical theoretical foundation. In this study, we develop an analysis that enables us to understand why PER cannot be effectively combined with continuous control algorithms and suggest novel modifications to PER to improve the empirical performance of the algorithm.

In actor-critic methods, a separate actor network is employed to choose continuous actions on the observed states. Therefore, when combined with PER, the actor and critic networks are trained with transitions that have large temporal-difference (TD) errors, where a large TD error implies that the critic has little knowledge and high uncertainty about the experience in terms of the bootstrapped expected future rewards [33]. However, we claim that actor networks cannot be effectively trained with experiences that the critic does not know their future returns well since their performance is assessed under the critic network. An intuitive analogy may be that it is infeasible to expect a student to learn a subject well if the teacher has little knowledge about it. Our main theoretical contributions in this work justify our claim that if an actor-critic algorithm is trained with a transition corresponding to a large TD error, the approximate policy gradient, i.e., computed under the Q-network, can significantly diverge from the actual gradient, i.e., computed under the optimal Q-function, for the

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

transition in interest or the subsequent transition. This finding can be used to improve the performance of PER by training the actor with different experiences and facilitating the design of novel experience prioritization methods. Discoveries of this study can be summarized as follows:

Actor networks should be trained with low TD error transitions. The critical implication of this finding is that the policy gradient, either stochastic or deterministic, that depends on the critic cannot be accurately computed using transitions on which the critic has high uncertainty. In particular, we find that a large TD error can correspond to a high Q-value estimation error for some transitions. Such error may cause the approximate policy gradient to diverge from the actual gradient computed under the optimal Q-function. To the best of our knowledge, this is the primary reason behind the poor performance of PER in standard off-policy actor-critic algorithms and we are the first to show it theoretically. Ultimately, this can only be overcome when the actor is trained with low TD error transitions in the TD error based prioritized sampling.

Actor and critic networks should be optimized with uniformly sampled transitions for a fraction of the batch size. Training actor and critic with completely different transitions, e.g., low and high TD error, violates the actor-critic theory since critic parameters always depend on the actor parameters as the actions selected by the actor are used in the updates [21]. Our empirical studies show that using a set of uniformly sampled transitions for a fraction of the batch size is extremely important in off-policy actor-critic training and ensures stability in learning.

Loss functions should be modified to prevent the outlier bias leakage in the prioritized sampling. While the combination of the latter two findings is theoretically and empirically favorable, prioritized [30] and uniform sampling should not be considered in isolation from the loss function as an outlier biased transitions may still leak during TD error based prioritized sampling [11]. Notably, we demonstrate that corrections to PER cannot reach their maximum potential unless the mean-squared error (MSE) in the Q-network training is corrected. Thus, we leverage the prominent results of Fujimoto et al. [11] in our modifications to PER.

We introduce Loss-Adjusted Approximate Actor Prioritized Experience Replay (LA3P), a novel prioritized sampling framework that adapts PER to continuous control by training the actor network with transitions that the critic has reliable knowledge of. Moreover, our algorithm considers the issues with stability and traditional actor-critic theory by not completely separating the actor and critic training and adjusting the loss functions accordingly. We evaluate LA3P on challenging OpenAI Gym [6] continuous control benchmarks and find that our method outperforms the competing PER correction algorithms by a large margin and obtains significant gains over PER and state-of-the-art in the majority of the tasks. All of our code and results are open-sourced and provided in the anonymous GitHub repository[1].

2 Related Work

Initial studies in experience prioritization originate from prioritized sweeping for value iteration [25, 1] to boost the learning speed and effectively use the computational resources. It is also utilized in modern applications of RL to perform importance sampling over the collected trajectories [31] and learning from demonstrations [17]. Prioritized Experience Replay [30] has been one of the most remarkable improvements to the DQN algorithm [24] and its successors [14, 35, 3], and is employed in many learning algorithms along with additional improvements [16, 18, 2]. Modifications on PER have also been proposed, e.g., prioritizing the sequences of transitions [12, 22, 7, 5] or optimization of the prioritization function [36]. A counterpart to the mentioned experience replay [20] approaches is determining which transitions to favor or forget [26]. Moreover, the effects originating from the composition and size of the experience replay buffers have also been studied [8, 9, 37, 19], along with prioritization in simple environments [23]. Finally, learning-based prioritization approaches through deep neural networks have been recently proposed to determine which experiences to sample, independent of the experience replay buffer composition and without deciding which transitions to store [36, 27]. In contrast to the mentioned approaches, our method focuses on the algorithmic drawbacks caused by applying PER to continuous control. We introduce corrections to PER in the context of the algorithm without proposing any alternative experience scoring scheme.

[1] `https://anonymous.4open.science/r/LA3P`

Lately, corrections to PER have been extensively investigated by Fujimoto et al. [11] and Oh et al. [28]. First, Fujimoto et al. [11] addressed that although the importance sampling is used, PER can still introduce bias when the MSE loss is used in the Q-network updates. To remedy this, the authors proposed a new loss function based on the Huber loss to be used with PER. Their algorithm, Loss Adjusted Prioritized (LAP) Experience Replay [11], has been shown to improve the performance of PER on several benchmarks. We broadly investigate LAP in later sections. Secondly, Oh et al. [28] claim that sampling from the replay buffer depending highly on the TD error (or Q-network's error) may be ineffective due to the under- or overestimation of the Q-values resulting from the deep function approximators and bootstrapping. For this reason, they proposed Model-Augmented PER (MaPER) [28] to learn auxiliary features driven from the components in model-based RL to calculate the scores of experiences. Ultimately, we include the theoretical results of Fujimoto et al. [11] in our methodology, while we also compare our method against LAP and MaPER in our empirical studies.

3 Technical Preliminaries

3.1 Deep Reinforcement Learning

This study considers the standard RL setup, represented by a finite Markov decision process consisting of a 5-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, P, \gamma)$, with state space $\mathcal{S}$, action space $\mathcal{A}$, a deterministic or stochastic reward function $\mathcal{R}$, the environment dynamics model P, and the discount factor $\gamma \geq 0$. The behavior of an RL agent is defined by its policy π, which can be deterministic if it maps states to unique actions $\pi : \mathcal{S} \rightarrow \mathcal{A}$, or stochastic if it maps states to action probabilities $\pi : \mathcal{S} \rightarrow p(\mathcal{A})$. The performance of a policy π is assessed under the action-value function (Q-function or critic) Q^π, which represents the expected sum of discounted rewards while following the policy π after performing the action a in state s: $Q^\pi(s,a) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0 = s, a_0 = a]$. The action-value function is determined through the Bellman equation [4]: $Q^\pi(s,a) = \mathbb{E}_{r,s' \sim P, a' \sim \pi}[r + \gamma Q^\pi(s', a')]$, where a' is the next action selected by the policy on the observed next state s'.

In deep RL, the critic is approximated by a deep neural network Q_θ with parameters θ, e.g., the DQN algorithm. Given a transition tuple $\tau = (s, a, r, s')$, the Q-network is trained by minimizing a loss $\mathcal{L}(\delta_\theta(\tau))$ on the temporal-difference (TD) error $\delta_\theta(\tau)$ corresponding to Q_θ [32], the difference between the output of Q_θ and learning target $y(\tau)$:

$$y(\tau) = r + \gamma Q_{\theta'}(s', a'), \tag{1}$$

$$\delta_\theta(\tau) = y(\tau) - Q_\theta(\tau). \tag{2}$$

Transitions $\tau \in \mathcal{B}$ are sampled through a sampling method from the experience replay buffer [20] that contains a previously collected set of experiences in the form of a batch of transitions $\mathcal{B}$. The target $y(\tau)$ in Equation (1) utilizes a separate target network with parameters θ' that maintains stability and fixed objective in learning the optimal Q-function. The target parameters are updated to copy the parameters θ after a number of learning steps. In each update step, the loss for the Q-network is averaged over the sampled batch of transitions $\mathcal{B}$: $\frac{1}{|\mathcal{B}|} \sum_{\tau \in \mathcal{B}} \mathcal{L}(\delta_\theta(\tau))$, where $|\mathcal{B}|$ is the number of transitions contained in $\mathcal{B}$.

In deep actor-critic methods, the policy is represented by the actor network π_ϕ, parameterized by ϕ, to choose continuous actions on the observed states since the maximum $\max_{\tilde{a}} Q(s, \tilde{a})$ to select actions is intractable due to an infinite number of possible actions. The policy is optimized with respect to the policy gradient $\nabla_\phi J(\phi)$, computed by a policy gradient technique, the loss of which is explicitly or implicitly based on maximizing the Q-value estimates of the Q-network.

3.2 Prioritized Experience Replay

Prioritized Experience Replay is a non-uniform sampling strategy for replay buffers in which transitions are sampled in proportion to their TD error. The primary reasoning for PER is that training on the highest error samples will yield the most significant performance improvement. PER introduces two modifications over the standard uniform sampling. First, a stochastic prioritization scheme is used. The motivation is that TD errors are updated only for replayed transitions. As a result, the initially high TD error transitions are updated more frequently, resulting in a greedy prioritization. Also, the noisy Q-value estimates increase the variance due to the greedy sampling. Therefore, overfitting is inevitable if one directly samples transitions proportional to their TD errors. To remedy

this, a probability value is assigned to each transition τ_i, proportional to the corresponding TD error $\delta_\theta(\tau_i)$, and set to the power of a hyper-parameter α to smooth out the extremes:

$$p(\tau_i) = \frac{|\delta_\theta(\tau_i)|^\alpha + \mu}{\sum_{j \in R}(|\delta_\theta(\tau_j)|^\alpha + \mu)}, \tag{3}$$

where R denotes the experience replay buffer and a small constant μ is added to avoid assigning zero probabilities to transitions; otherwise, they would not be sampled again. This is required since the most recent value of a transition's TD error is approximated by the TD error when it was last sampled.

Second, favoring large TD error transitions with the stochastic prioritization shifts the distribution of s' to $\mathbb{E}_{s'}[Q(s', a')]$. This can be corrected through importance sampling with ratios $w(\tau_i)$:

$$\hat{w}(\tau_i) = \left(\frac{1}{|R|} \cdot \frac{1}{p(\tau_i)}\right)^\beta, \quad w(\tau_i) = \frac{\hat{w}(\tau_i)}{\max_j \hat{w}(\tau_j)}; \tag{4}$$

$$\mathcal{L}_{\text{PER}}(\delta_\theta(\tau_i)) = w(\tau_i)\mathcal{L}_{\text{MSE}}(\delta_\theta(\tau_i)), \tag{5}$$

where $|R|$ is the total number of transitions in the replay buffer, $\mathcal{L}_{\text{MSE}}(\delta_\theta(\tau_i)) = 0.5\delta_\theta(\tau_i)^2$, and the hyper-parameter β is used to smooth out the high variance induced by the importance sampling weights. With the latter equation, the distribution shift is corrected such that the effect of high priorities is reduced by using a ratio between uniform sampling with probability $\frac{1}{|R|}$ and the ratio in Equation (3). Lastly, the β value is annealed from a pre-defined initial value β_0 to 1, to eliminate the bias introduced by the distributional shift.

4 Prioritized Sampling in Actor-Critic Algorithms

We start by building the theoretical foundations for the performance degradation of the TD error based prioritized sampling [30] in actor-critic methods. First, we demonstrate that there exist transitions such that the associated TD errors are directly proportional to the Q-value estimation errors, where such Q-value estimates are used to compute the policy gradient. Then, using our theoretical implications, we show that the approximate policy gradient computed under the Q-network diverges from the actual gradient computed under the optimal Q-function if the policy is optimized using transitions with large TD errors. In addition to our theoretical investigation, we address the existing problems shown Fujimoto et al. [11] that explain the poor performance of PER with standard off-policy algorithms. All proofs are provided in Appendix A.

Lemma 1. *If δ_θ is the temporal-difference error associated with the critic network Q_θ, then there exists a transition tuple $\tau_t = (s_t, a_t, r_t, s_{t+1})$ with $\delta_\theta(\tau_t) \neq 0$ such that the absolute temporal-difference error on τ_t is directly proportional to the absolute estimation error on at least τ_t or τ_{t+1}:*

$$|\delta_\theta(\tau_t)| \propto |Q_\theta(s_i, a_i) - Q^\pi(s_i, a_i)|; \quad i = t \vee (t+1), \tag{6}$$

where $Q^\pi(s_i, a_i)$ is the actual Q-value of the state-action pair (s_i, a_i) while following the policy π.

The estimation error in RL with function approximation is often caused by using function approximators and bootstrapping, such as in the Q-learning algorithm [33]. As the estimation error is not usually distinguished into the function approximation and bootstrapping, it cannot be deduced that there is *always* a direct proportionality between estimation error and TD error. In addition, the TD-learning [32] can be a poor estimate in some conditions, such as when the rewards are noisy [33]. The TD error is a measure of how unexpected or surprising a transition is, not the estimation accuracy [25]. Therefore, Lemma 1 has to be made to show that the correlation between estimation and TD error exists for *some* state-action pairs. Next, we leverage our latter result to explain why policy optimization cannot be effectively performed using transitions with large TD errors.

Theorem 1. *Let τ_i be a transition such that Lemma 1 is satisfied. Then, if $\delta_\theta(\tau_i) \neq 0$, the following relation holds:*

$$|\delta_\theta(\tau_i)| \propto |\nabla_\phi J(\phi(\tau_j)) - \nabla_\phi J(\phi_{\textit{true}}(\tau_j))|; \quad j = i \vee (i+1), \tag{7}$$

where $\nabla_\phi J(\phi(\tau_j))$ and $\nabla_\phi J(\phi_{\textit{true}}(\tau_j))$ are the resulting policy gradients corresponding to τ_j if computed under the Q-network Q_θ and optimal Q-function Q^π, respectively.

Theorem 1 states that if the actor network is optimized with a transition corresponding to a large TD error, the resulting approximate policy gradient at the current or subsequent step may diverge from the actual gradient. We formally state this finding in Corollary 1.

Corollary 1. *If the temporal-difference error of a transition increases, the approximate policy gradient computed by any policy gradient algorithm with respect to the Q-network can diverge from the actual gradient computed under the optimal Q-function for the current or subsequent transition.*

This forms an essential ingredient in the degraded performance of the TD error based prioritized sampling when an actor network is employed. We now address a recent finding that explains a complement to the poor performance of PER in continuous control. As discussed, prioritizing transitions with large TD error through stochastic sampling shifts the distribution of s' to $\mathbb{E}_{s'}[Q(s', a')]$. Therefore, this induced bias is corrected by importance sampling, as expressed in Equation (5). However, Fujimoto et al. [11] argued that PER does not entirely eliminate the bias and may favor outliers when combined with the MSE loss in the Q-network updates, which is one of the reasons for the possible performance degradation of PER. To overcome this, Fujimoto et al. [11] replaced the MSE loss with the commonly used Huber loss with $\kappa = 1$:

$$\mathcal{L}_{\text{Huber}}(\delta_\theta(\tau_i)) = \begin{cases} 0.5\delta_\theta(\tau_i)^2 & \text{if } |\delta_\theta(\tau_i)| \leq \kappa, \\ |\delta_\theta(\tau_i)| & \text{otherwise.} \end{cases} \tag{8}$$

In addition to the above loss function, the following modified stochastic prioritization scheme is used:

$$p(\tau_i) = \frac{\max(|\delta_\theta(\tau_i)|^\alpha, 1)}{\sum_{j \in R} \max(|\delta_\theta(\tau_j)|^\alpha, 1)}. \tag{9}$$

Note that the clipping reduces the likelihood of dead transitions when $p(\tau_i) \approx 0$, which eliminates the need for the μ parameter.

The results presented by Fujimoto et al. [11] form a complement to our theoretical conclusions for the poor performance of PER in continuous control. There may be another justification, such as inaccurate Q-value estimates. However, such reasons are not PER-dependent and are induced by the DQN variants such as [14]. Therefore, to the best of our knowledge, we believe that Corollary 1 and results of Fujimoto et al. [11] establish the basis for the algorithmic drawbacks of PER in continuous action spaces.

5 Adaptation of Prioritized Experience Replay to Actor-Critic Algorithms

5.1 Inverse Sampling for the Actor Network

To remedy the mentioned issues of TD error based prioritized sampling in actor-critic algorithms, we introduce a set of novel modifications to vanilla PER. We start with Corollary 1, which we overcome by optimizing the actor network with transitions that have small TD errors. For this, as in the PER algorithm, *proportional prioritization* can be used. An efficient and the most popular implementation of proportional prioritization is based on a "sum-tree" data structure [30]. Thus, an instinctive approach to sample transitions with a probability inversely proportional to the TD error accommodates creating a new sum tree containing the priorities' global inverse. Although priorities in vanilla PER are updated for every training step through the sum tree data structure, inverse sampling for actor updates requires creating a new sum tree prior to training. In the learning step, we calculate the priorities stored in the new sum tree as follows:

$$I \sim \tilde{p}(\tau_i) = \frac{p_{\text{max}}}{p(\tau_i)} = \max_i \left(\frac{\max(|\delta_\theta(\tau_i)|^\alpha, 1)}{\sum_{j \in R} \max(|\delta_\theta(\tau_j)|^\alpha, 1)} \right) \cdot \frac{\sum_{j \in R} \max(|\delta_\theta(\tau_j)|^\alpha, 1)}{\max(|\delta_\theta(\tau_i)|^\alpha, 1)}, \tag{10}$$

where $p(\tau_i)$ is the priority of the ith transition and $p_{\max}$ is the maximum of the stored transitions' priorities. Notice that Equation (10) does not alter proportional prioritization. The relative proportions, e.g., the largest over the smallest, do not change as we take the inverse by multiplication. As mentioned, MSE with PER still induces varying biases that may favor outlier transitions. Hence, we adopt the prioritization scheme of LAP, expressed by Equation (9), in Equation (10).

This forms the fundamental component of our approach. To couple with the prioritization scheme of LAP, we employ the Huber loss with $\kappa = 1$ defined in Equation (8) for the Q-network updates,

similar to the LAP algorithm. Therefore, at every training step, Q-network and priorities are updated respectively as:

$$I \sim p(\tau_i) = \frac{\max(|\delta_\theta(\tau_i)|^\alpha, 1)}{\sum_{j \in R} \max(|\delta_\theta(\tau_j)|^\alpha, 1)}; \tag{11}$$

$$\theta \leftarrow \theta - \eta \cdot \frac{1}{|I|} \sum_{i \in I} \nabla_\theta \mathcal{L}_{\text{Huber}}(\delta_\theta(\tau_i)), \tag{12}$$

$$p(\tau_i) \leftarrow \max(|\delta_\theta(\tau_i)|^\alpha, 1) \text{ for } i \in I, \tag{13}$$

where I are indices of the sampled batch of prioritized transitions and η is the learning rate.

5.2 Optimizing the Actor and Critic with a Shared Set of Transitions

Training the actor and critic networks with entirely different transitions can violate the actor-critic theory. In general, features used by the critic network depend on the actor parameters and policy gradient since the actor determines the actions, and these actions lead to the observed state space [21]. A trivial corollary is that if the critic is updated by a set of features that lie in a state-action space in which the actor is never optimized, a substantial instability may occur since the transitions used by the critic are processed through the actor and the actor never sees those transitions [21]. Therefore, the reliability of critic's action evaluations might become questionable.

Intuitively, this situation can occur when prioritized and inverse prioritized sampling are used for the critic and actor networks, respectively, since they may never be optimized with the same transitions. This can be the case when the TD error of the critic's samples are not decreased such that the actor never sees them. We indicated that there is not always a direct correlation between TD and estimation errors. Hence, some of the transitions may initially have low TD errors. If the actor is optimized with respect to these low TD error transitions throughout the learning and the Q-network only focuses on the remaining large TD error transitions, samples used in the actor and critic training might not be the same. Although this remains a little possibility, we nevertheless overcome this by updating the actor and critic networks through a set of shared transitions, being a fraction of the number of samples used in each learning step. However, we cannot know the value of such a fraction, and we introduce it here as the hyper-parameter λ.

How to choose the set of shared transitions? In the context of TD error based prioritization, we have the alternatives of: (i) transitions with large TD error, (ii) transitions with small TD error, and (iii) uniformly sampled transitions. We eliminate the first as it contradicts with Corollary 1. Moreover, learning from experiences with small TD errors can be beneficial. Nonetheless, they decrease the sampling efficiency and waste resources since the Q-network has little to learn from small TD error transitions, i.e., the Q-network loss would be small. Thus, the latter two alternatives imply that uniform sampling for the set of shared transitions remains the only choice. Although large TD error transitions might be included in the uniformly sampled mini-batch, their effects are reduced due to averaging in the mini-batch learning. Furthermore, relying on random sampling can include transitions that could not be sampled by prioritized and inverse prioritized sampling. We may also utilize transitions with average magnitudes of TD errors. Nevertheless, uniform sampling already corresponds to transitions with mean TD error in the expectation.

As discussed, the combination of Huber loss ($\kappa = 1$) with the prioritized sampling can eliminate the outlier bias in the LAP algorithm. Fujimoto et al. [11] also introduced the mirrored loss function of LAP, with an equivalent expected gradient, for uniform sampling from the experience replay buffer. To observe the same benefits of LAP also in the uniform sampling counterpart, its mirrored loss function, Prioritized Approximate Loss (PAL) [11], should be employed instead of MSE. The PAL function is expressed as:

$$\xi = \frac{\sum_{j \in R} \max(|\delta_\theta(\tau_j)|^\alpha, 1)}{|R|}; \tag{14}$$

$$\mathcal{L}_{\text{PAL}}(\delta_\theta(\tau_i)) = \frac{1}{\xi} \begin{cases} 0.5\delta_\theta(\tau_i)^2 & \text{if } |\delta_\theta(\tau_i)| \leq 1, \\ \frac{|\delta_\theta(\tau_i)|^{1+\alpha}}{1+\alpha} & \text{otherwise.} \end{cases} \tag{15}$$

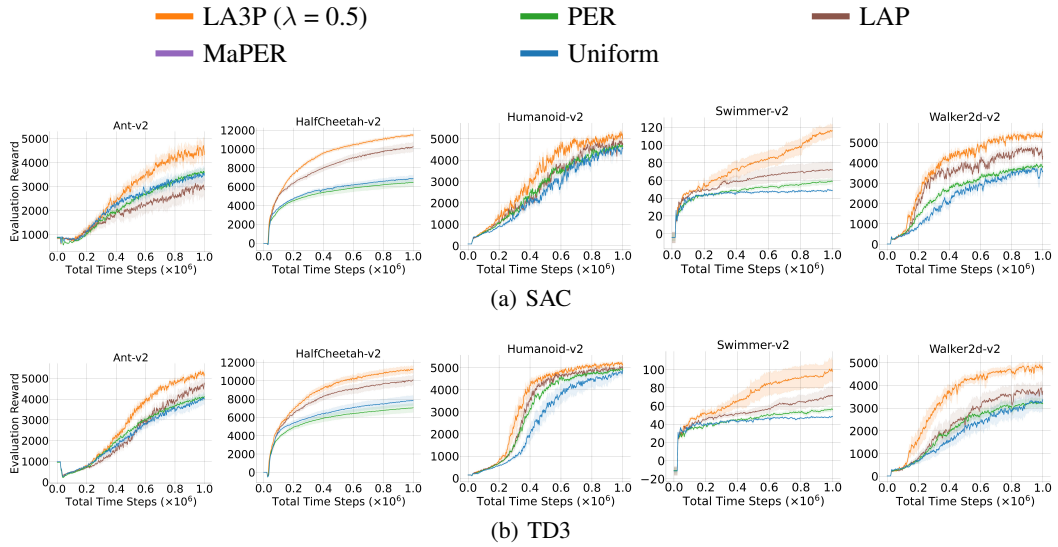

Figure 1: Learning curves for the set of MuJoCo continuous control tasks under the SAC and TD3 algorithms. Curves are averaged over 10 trials, where the shaded region represents a 95% confidence interval over the trials.

Having the latter component included, this forms our PER correction algorithm, **L**oss-**A**djusted **A**pproximate **A**ctor **P**rioritized Experience Replay (LA3P), which we extensively describe its structure in Appendix B. Lastly, we perform a complexity analysis for LA3P in Appendix C.

6 Experiments

With all the mentioned concepts combined, we investigate to what extent our prioritization framework can improve the performance of off-policy actor-critic methods in continuous control. Thus, we perform experiments to evaluate the effectiveness of LA3P on the standard suite of MuJoCo [34] and Box2D [29] continuous control tasks interfaced by OpenAI Gym. We combine our method with the state-of-the-art off-policy actor-critic algorithms, Twin Delayed Deep Deterministic Policy Gradient (TD3) [10] and Soft Actor-Critic (SAC) [13], which we benchmark against uniform sampling, PER, and the PER correction methods of LAP and MaPER. Exact hyper-parameter settings, architecture, and implementation are broadly explained in Appendix D.

6.1 Comparative Evaluation

Learning curves for the set of OpenAI Gym continuous control benchmarks are reported in Figure 1 for the SAC and TD3 algorithms, and results in additional environments are given in Appendix E.1. We also report the average of the last ten evaluation returns, i.e., the level where the algorithms converge, in Appendix E.2. Empirical complexity analysis is provided in Appendix F. From our evaluation results, we observe that LA3P matches or outperforms the competing approaches in all tasks and baseline off-policy actor-critic algorithms tested. However, for the BipedalWalker and LunarLanderContinuous environments under the SAC algorithm, we find that LAP obtains slightly larger rewards than our method. Nevertheless, these scores can be practically counted as the same, and as the learning curves show, LA3P could attain faster convergence. In trivial environments such as LunarLanderContinuous, a minor improvement is achieved. However, in the Swimmer environment where no algorithm could converge, the performance improvement offered by our modifications becomes more prominent. Furthermore, we also observe a substantial improvement by LA3P in the HalfCheetah environment over the prior approaches. As HalfCheetah and Swimmer are regarded as "stable", i.e., episodes terminate only if the pre-specified number of time steps is reached, they require long horizons to be simulated. Therefore, as stated by Fujimoto et al. [11], the benefit of a corrected prioritization scheme is more observable in environments with longer horizons.

Additionally, we confirm previous empirical studies, e.g., [11, 28, 27], which found that PER provides no benefits when added to off-policy continuous control algorithms, and performance is usually degraded. While this is attributed to the use of MSE by Fujimoto et al. [11], prioritization with corrected loss function, i.e., LAP, appears to have little impact in Ant, Hopper, and Walker2d, compared to LA3P. In fact, the SAC algorithm is underperformed in Ant and Hopper by the LAP algorithm. This result is consistent with our theoretical analysis made in Corollary 1, which shows that optimizing the actor network with transitions corresponding to large TD errors can cause the approximate policy gradient to diverge from the one computed under the optimal Q-function. In these environments, learning curves demonstrate that the performance gain offered by LA3P primarily comes from the inverse prioritized sampling for the actor network. This suggests that the inverse sampling in the LA3P framework plays a more significant role than the employed LAP and PAL functions. Hence, the combined solution of LA3P, inverse sampling with corrected loss functions, is superior.

Finally, we notice that the performance of MaPER is not promising as it acquires slight improvements over PER. We attribute such a poor performance mainly to the model prediction structure of the algorithm. As we previously discussed, the MaPER algorithm focuses on new learnable features driven by the components in model-based RL to calculate the scores on experiences since critic networks often under- or overestimate Q-values. However, the Clipped Double Q-learning algorithm proposed by Fujimoto et al. [10] already solves the issues with inaccurate Q-value estimates, which is already employed in SAC and TD3. Therefore, we conclude that the main drawback of PER is not the inaccurate Q-value estimates used in the priority calculations but the biased loss function and training the actor network with large TD error transitions. In addition, the model prediction module in MaPER decreases the convergence rate yet, brings notable stability to the learning. Nonetheless, the resulting performance is not considerable. Consequently, we believe that LA3P is a preferable and comprehensive way of overcoming the underlying issues of PER in continuous action domains.

6.2 Ablation Studies

To better understand the contribution of each component in LA3P, we conduct an ablation study. The LA3P algorithm introduces several modifications to PER. In summary, LA3P consists of: (i) inverse sampling for the actor, (ii) uniform sampling for the actor and critic networks to share a set of transitions, (iii) the LAP function applied to the prioritized transitions, (iv) the PAL function applied to the uniformly sampled transitions.

We evaluate and discuss the resulting performances when removing each of these components. In addition, we test the performance of LA3P when the shared transitions are low TD error experiences instead of uniformly sampled ones to demonstrate the mentioned decreased data efficiency. Finally, we perform a sensitivity analysis for the λ parameter. We do not remove the inverse sampling as it is the backbone of our algorithm, that is, eliminating the inverse sampling for the actor network would not relate to any of the modifications introduced by this work as it would be just a mixture of uniform and prioritized sampling. Moreover, we choose four challenging environments with different characteristics for a comprehensive inference. As described by Henderson et al. [15] and Fujimoto et al. [11], we consider the stable environment of HalfCheetah, the unstable environment of Walker2d, and the high dimensional and most challenging environments of Ant and Humanoid.

Table 1 presents the average of the last ten evaluation returns over ten trials, i.e., the level where the tested settings converge, for our ablation studies and sensitivity analysis, and the corresponding learning curves are depicted in Appendix G. The same experimental setup is used to perform the ablation studies, and $\lambda = 0.5$ is used for all experiments unless otherwise stated. Note that $\lambda = 0.0$ yields the LA3P setting without a shared set of transitions, and $\lambda = 1.0$ corresponds to uniform sampling, the results of which were already provided.

First, we deduce that the set of shared transitions is the most crucial component of our framework. Independently training the actor and critic networks violate their correlation as the actor is optimized by maximizing the Q-values estimated by the Q-network, and the Q-network is trained using the actions selected by the actor. Thus, they should not be separated in training, and we empirically verify our previous discussion on transition sharing. Although the LAP and PAL functions apply the same number of transitions in each update step, i.e., $\lambda = 0.5$, we observe that the contribution of LAP is more significant than PAL. As discussed in our comparative evaluations, the performance improvement by LA3P largely relies on inverse sampling for the actor network. As expected,

Table 1: Average return over the last 10 evaluations over 10 trials of 1 million time steps, comparing ablation over LA3P under low TD error shared transitions, LA3P without the LAP function, LA3P without the PAL function, LA3P without the shared set of transitions, and LA3P under $\lambda = \{0.1, 0.3, 0.5, 0.7, 0.9\}$. $\pm$ captures a 95% confidence interval over the trials. Bold values represent the maximum under each environment.

Setting	**Ant**	**HalfCheetah**	**Humanoid**	**Walker2d**
Low TD-error	**3485.1 $\pm$ 946.0**	**10992.1 $\pm$ 485.0**	**3938.1 $\pm$ 1418.1**	**4438.3 $\pm$ 398.5**
w/o LAP	3408.6 $\pm$ 596.7	4580.3 $\pm$ 257.9	3585.1 $\pm$ 950.3	3262.5 $\pm$ 388.6
w/o PAL	3975.3 $\pm$ 1207.9	7560.5 $\pm$ 774.6	4879.4 $\pm$ 352.0	4543.9 $\pm$ 528.7
w/o Uniform	4431.9 $\pm$ 787.8	10483.1 $\pm$ 600.5	5058.7 $\pm$ 183.9	4254.6 $\pm$ 390.6
$\lambda = 0.1$	3768.7 $\pm$ 1079.6	11203.8 $\pm$ 576.1	4695.5 $\pm$ 338.4	4562.8 $\pm$ 406.0
$\lambda = 0.3$	4903.3 $\pm$ 467.5	10460.5 $\pm$ 948.6	4528.3 $\pm$ 1148.4	4425.1 $\pm$ 448.9
$\lambda = 0.5$	**5197.5 $\pm$ 377.3**	**11225.1 $\pm$ 811.6**	**5131.1 $\pm$ 250.8**	**4776.7 $\pm$ 424.5**
$\lambda = 0.7$	4384.0 $\pm$ 924.9	10656.9 $\pm$ 750.9	4874.5 $\pm$ 267.6	4253.8 $\pm$ 905.0
$\lambda = 0.9$	3882.1 $\pm$ 1157.6	10547.8 $\pm$ 891.0	3757.9 $\pm$ 1403.4	4546.0 $\pm$ 593.7

correcting the prioritization in inverse sampling for the actor network through the LAP approach, i.e., Equation (10), is more crucial than correcting the loss by PAL for the uniformly sampled batch. Lastly, we infer that using low TD error transitions instead of uniformly sampled ones substantially degrades the performance. Although this setting would seem to be a reasonable choice at first glance, the data efficiency considerably decreases as the Q-network repeatedly trains with transitions that it has already learned well. In the expectation, the uniformly sampled batch of transitions corresponds to an intermediate TD error value compared to the transitions contained in the entire replay buffer. As we experimentally show, this may benefit both the actor and critic networks since inverse prioritized and prioritized sampling may not include transitions with intermediate TD error values.

Our sensitivity analysis on the λ parameter suggests that $\lambda = 0.5$ produces the best results by a notable margin in all environments. As λ decreases, the correlation between the actor and critic networks starts to be ignored, and the performance drops. In contrast, the larger λ values yield the performance to converge to that of uniform sampling. Hence, we believe the introduced framework does not require intensive hyper-parameter tuning, and $\lambda = 0.5$ can apply to many tasks. Overall, it is shown by our ablation studies that our framework improves over the baseline actor-critic algorithms due to the structure of the introduced method rather than unintended consequences or any exhaustive hyper-parameter tuning.

7 Conclusion

In this paper, we build the theoretical foundations behind the poor empirical performance of a widely known experience replay sampling algorithm, Prioritized Experience Replay (PER) [30], for controlling continuous systems. To achieve this, we first show that training actor networks with large TD errors in the PER algorithm may cause the approximate policy gradient computed under the Q-network to diverge from the one computed under the optimal Q-function. This result suggests that even if the biased loss function in the PER algorithm is corrected, optimizing the actor network with low TD error transitions can significantly increase performance. This enables us to comprehend PER's poor performance in more detail when applied to continuous control algorithms.

However, training actor and critic networks with different transitions can violate their dependence as the actor tries to maximize the Q-value estimated by the Q-network, and Q-network computes its loss based on the actions selected by the actor. This allows us to develop a novel framework, Loss Adjusted Approximate Actor Prioritized Experience Replay (LA3P), which mixes training with uniformly sampled, low, and high TD error transitions. The introduced approach also accounts for the previous findings of Fujimoto et al. [11], which practically eliminate the outlier bias introduced by the combination of mean-squared error with PER. We test LA3P on standard deep reinforcement learning benchmarks in MuJoCo and Box2D and demonstrate that it substantially outperforms the competing methods and improves the state-of-the-art.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We clearly describe the problem of interest and our contributions to solving it in both the abstract and introduction sections. For example, on the second page of Section 1, we summarized the discoveries of our study across three paragraphs.
   (b) Did you describe the limitations of your work? [Yes] We repeatedly indicate the limitations of our work throughout the paper. For instance, in Section 1, we stated that our theoretical results are limited to the off-policy actor-critic methods, and the introduced method solves the problems in the PER algorithm using only the TD error based prioritization scheme.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] All authors read and ensured that the presented work conforms to the ethics review guidelines.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Lemma 1 and Theorem 1 completely state the full set of assumptions.
   (b) Did you include complete proofs of all theoretical results? [Yes] All proofs are provided in Appendix A.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] All of the code used to produce the reported results are open-sourced in the anonymous GitHub repository[1]. In addition, the presented results and relevant instructions for reproduction are also provided in the repository. In case of acceptance, the repository will be deanonymized.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] All of the training details are precisely explained in Appendix D.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] As it can be observed in Figure 1, and figures in Appendix E.1 and Appendix G, we report the 95% confidence intervals over the tested ten random seeds.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We include the computing infrastructure used for running our experiments both in our GitHub repository[1] and Appendix F.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] The used code and simulators are either cited or URLs are given when they are first mentioned, e.g., see Appendix D.
   (b) Did you mention the license of the assets? [N/A] All of the used assets are already open-sourced, and URLs are provided.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See Appendix D.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] All of the used data are publicly available.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

References

[1] David Andre, Nir Friedman, and Ronald Parr. Generalized prioritized sweeping. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. MIT Press, 1997. URL https://proceedings.neurips.cc/paper/1997/file/7b5b23f4aadf9513306bcd59afb6e4c9-Paper.pdf.

[2] Gabriel Barth-Maron, Matthew W. Hoffman, David Budden, Will Dabney, Dan Horgan, Dhruva TB, Alistair Muldal, Nicolas Heess, and Timothy Lillicrap. Distributional policy gradients. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SyZipzbCb.

[3] Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 449–458. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/bellemare17a.html.

[4] Richard Ernest Bellman. *Dynamic Programming*. Dover Publications, Inc., USA, 2003. ISBN 0486428095.

[5] Marc Brittain, Josh Bertram, Xuxi Yang, and Peng Wei. Prioritized sequence experience replay, 2019. URL https://arxiv.org/abs/1905.12726.

[6] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *CoRR*, abs/1606.01540, 2016. URL http://arxiv.org/abs/1606.01540.

[7] Brett Daley and Christopher Amato. Reconciling λ -returns with experience replay. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/9f396fe44e7c05c16873b05ec425cbad-Paper.pdf.

[8] Tim De Bruin, Jens Kober, Karl Tuyls, and Robert Babuška. The importance of experience replay database composition in deep reinforcement learning.

[9] Tim de Bruin, Jens Kober, Karl Tuyls, and Robert Babuška. Improved deep reinforcement learning for robotics through distribution-based experience retention. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3947–3952, 2016. doi: 10.1109/IROS.2016.7759581.

[10] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1587–1596, Stockholmsmässan, Stockholm SWEDEN, 10–15 Jul 2018. PMLR. URL https://proceedings.mlr.press/v80/fujimoto18a.html.

[11] Scott Fujimoto, David Meger, and Doina Precup. An equivalence between loss functions and non-uniform sampling in experience replay. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 14219–14230. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/a3bf6e4db673b6449c2f7d13ee6ec9c0-Paper.pdf.

[12] Audrunas Gruslys, Will Dabney, Mohammad Gheshlaghi Azar, Bilal Piot, Marc Bellemare, and Remi Munos. The reactor: A fast and sample-efficient actor-critic agent for reinforcement learning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkHVZWZAZ.

[13] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine*

Learning, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

[14] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 2094–2100, Phoenix, Arizona, 2016. AAAI Press.

[15] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18, New Orleans, Louisiana, USA, 2018. AAAI Press. ISBN 978-1-57735-800-8.

[16] 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. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. URL https://ojs.aaai.org/index.php/AAAI/article/view/11796.

[17] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, Gabriel Dulac-Arnold, John Agapiou, Joel Leibo, and Audrunas Gruslys. Deep q-learning from demonstrations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. URL https://ojs.aaai.org/index.php/AAAI/article/view/11757.

[18] Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado van Hasselt, and David Silver. Distributed prioritized experience replay. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1Dy---0Z.

[19] David Isele and Akansel Cosgun. Selective experience replay for lifelong learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. URL https://ojs.aaai.org/index.php/AAAI/article/view/11595.

[20] Long ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. In *Machine Learning*, pages 293–321, 1992.

[21] Vijay Konda and John Tsitsiklis. Actor-critic algorithms. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[22] Su Young Lee, Choi Sungik, and Sae-Young Chung. Sample-efficient deep reinforcement learning via episodic backward update. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/e6d8545daa42d5ced125a4bf747b3688-Paper.pdf.

[23] Ruishan Liu and James Zou. The effects of memory replay in reinforcement learning. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 478–485, 2018. doi: 10.1109/ALLERTON.2018.8636075.

[24] 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*, 518(7540):529–533, Feb 2015. ISSN 1476-4687. doi: 10.1038/nature14236. URL https://doi.org/10.1038/nature14236.

[25] Andrew W. Moore and Christopher G. Atkeson. Prioritized sweeping: Reinforcement learning with less data and less time. *Machine Learning*, 13(1):103–130, Oct 1993. ISSN 1573-0565. doi: 10.1007/BF00993104. URL https://doi.org/10.1007/BF00993104.

[26] Guido Novati and Petros Koumoutsakos. Remember and forget for experience replay. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4851–4860. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/novati19a.html.

[27] Youngmin Oh, Kimin Lee, Jinwoo Shin, Eunho Yang, and Sung Ju Hwang. Learning to sample with local and global contexts in experience replay buffer. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=gJYlaqL8i8.

[28] Youngmin Oh, Jinwoo Shin, Eunho Yang, and Sung Ju Hwang. Model-augmented prioritized experience replay. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=WuEiafqdy9H.

[29] Ian Parberry. *Introduction to Game Physics with Box2D*. CRC Press, Inc., USA, 1st edition, 2013. ISBN 1466565764.

[30] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay, 2015. URL http://arxiv.org/abs/1511.05952. cite arxiv:1511.05952Comment: Published at ICLR 2016.

[31] Matthew Schlegel, Wesley Chung, Daniel Graves, Jian Qian, and Martha White. *Importance Resampling for Off-Policy Prediction*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[32] Richard Sutton. Learning to predict by the method of temporal differences. *Machine Learning*, 3:9–44, 08 1988. doi: 10.1007/BF00115009.

[33] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018. ISBN 0262039249.

[34] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

[35] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1995–2003, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/wangf16.html.

[36] Daochen Zha, Kwei-Herng Lai, Kaixiong Zhou, and Xia Hu. Experience replay optimization. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4243–4249. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/589. URL https://doi.org/10.24963/ijcai.2019/589.

[37] Shangtong Zhang and Richard S. Sutton. A deeper look at experience replay. In *NIPS 2017 Deep Reinforcement Learning Symposium*, 2017.