# Stratified Experience Replay: Correcting Multiplicity Bias in Off-Policy Reinforcement Learning

Extended Abstract

Brett Daley
Northeastern University
Boston, MA, USA
b.daley@northeastern.edu

Cameron Hickert
Harvard University
Cambridge, MA, USA
cameron_hickert@hks.harvard.edu

Christopher Amato
Northeastern University
Boston, MA, USA
c.amato@northeastern.edu

## KEYWORDS

Deep reinforcement learning, Experience replay

**ACM Reference Format:**
Brett Daley, Cameron Hickert, and Christopher Amato. 2021. Stratified Experience Replay: Correcting Multiplicity Bias in Off-Policy Reinforcement Learning: Extended Abstract. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 3 pages.

## 1 INTRODUCTION

Deep Reinforcement Learning (RL) methods rely on experience replay [9] to approximate the minibatched supervised learning setting; however, unlike supervised learning where access to lots of training data is crucial to generalization, replay-based deep RL appears to struggle in the presence of extraneous data. Recent works have shown that the performance of Deep Q-Network (DQN) [11] degrades when its replay memory becomes too large [4, 10, 17].

This suggests that outdated experiences somehow impact the performance of deep RL, which should not be the case for off-policy methods like DQN. Consequently, we re-examine the motivation for sampling *uniformly* over a replay memory, and find that it may be flawed when using function approximation. We show that—despite conventional wisdom—sampling from the uniform distribution does not yield uncorrelated training samples and therefore biases gradients during training. Our theory prescribes a special non-uniform distribution to cancel this effect, and we propose a stratified sampling scheme to efficiently implement it (see Figure 1).

## 2 MOTIVATION

We begin by showing how bias arises under experience replay with function approximation by comparing Q-Learning [16] with its deep analog, DQN [11]. We model the environment as a Markov Decision Process (MDP) of the standard form $(\mathcal{S}, \mathcal{A}, T, R)$ [14].

Upon taking action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$ and observing the resulting state $s' \in \mathcal{S}$, Q-Learning conducts an update on an entry $Q(s, a)$ of its lookup table. Define the temporal-difference error as $\delta(s, a, s') = R(s, a, s') + \gamma \max_{a' \in \mathcal{A}} Q(s', a') - Q(s, a)$ with discount factor $\gamma \in [0, 1]$. Since this particular error has probability $T(s, a, s') = \Pr(s' \mid s, a)$ of occurring, the *expected* Q-Learning update can be computed:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \sum_{s' \in \mathcal{S}} \Pr(s' \mid s, a)\delta(s, a, s') \tag{1}$$

*Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.*


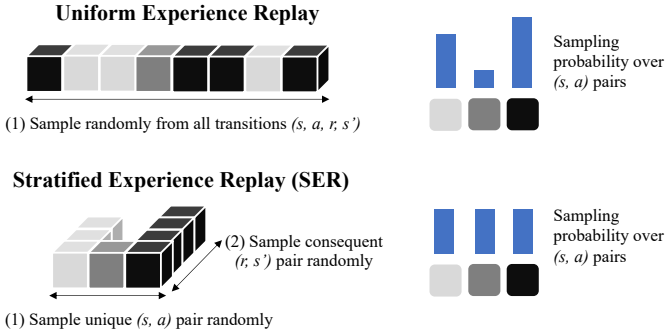


**Figure 1: A graphical comparison of uniform (top) and stratified (bottom) sampling strategies.**

**Data Structure 1** Stratified Replay Memory

**Initialize** array $D$ of size $N$, hash table $H$, integer $i = 0$

**procedure** INSERT($s$, $a$, $r$, $s'$)
  **if** $D$ is full **then**
    Get transition $(s_i, a_i, r_i, s_i')$ from $D[i]$
    Pop queue $H[(s_i, a_i)]$; if now empty, delete key $(s_i, a_i)$
  **end if**
  If $(s, a) \notin H$, then $H[(s, a)] \leftarrow$ *empty queue*
  Push $i$ onto queue $H[(s, a)]$
  $D[i] \leftarrow (s, a, r, s')$; $i \leftarrow (i + 1) \bmod N$
**end procedure**

**function** SAMPLE( )
  Sample state-action pair $(s, a)$ uniformly from the keys of $H$
  Sample integer $j$ uniformly from queue $H[(s, a)]$
  **return** transition $(s_j, a_j, r_j, s_j')$ from $D[j]$
**end function**

where $\alpha \in [0, 1]$ is the learning rate. Importantly, the expected Q-Learning update is independent of the visitation frequency of the state-action pair $(s, a)$ as long as its probability of occurrence is nonzero.

Contrast this with DQN, which replaces the tabular lookup $Q(s, a)$ with a parametric function $Q(s, a; \theta)$ that is trained via stochastic gradient descent over a dataset of past experiences. To facilitate our analysis, consider the theoretical case where DQN's replay memory $D$ has unlimited capacity and the agent executes a fixed behavior policy $\mu$ for an infinite duration before training. We can deduce that a sample drawn uniformly from $D$ will have probability

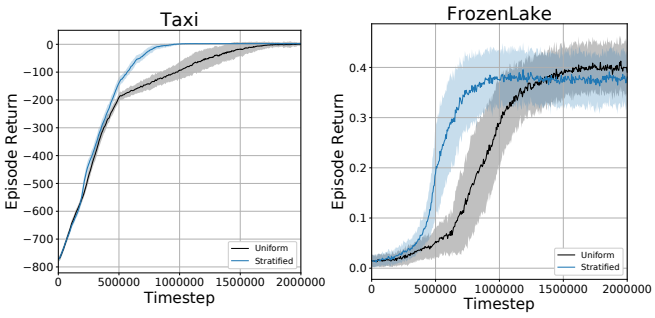


**Figure 2: SER performance compared against a uniform baseline on two environments, averaged over 100 trials.**

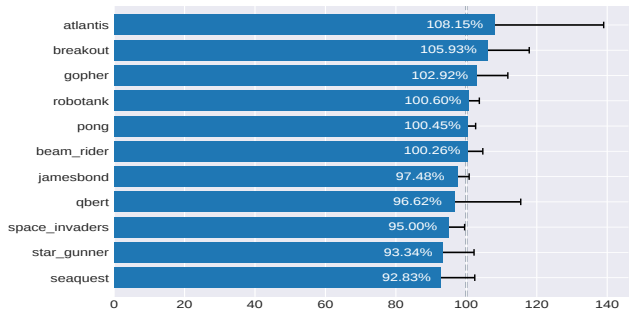


**Figure 3: Average episode score of SER throughout training on 11 Atari games, relative to that of the uniform baseline, i.e.** $100 \times$ **(stratified − random) / (uniform − random).**

$\Pr(s, a, r, s') = \Pr(s' \mid s, a) \Pr(s, a)$.[1] Define the temporal-difference error as $\delta(s, a, s') = R(s, a, s') + \gamma \max_{a' \in \mathcal{A}} Q(s', a'; \theta^-) - Q(s, a; \theta)$ where $\theta^-$ is a time-delayed copy of $\theta$ that helps stabilize training. The *expected* DQN update can likewise be computed:

$$\theta \leftarrow \theta + \alpha \Pr(s, a) \sum_{s' \in \mathcal{S}} \Pr(s' \mid s, a)\, \delta(s, a, s')\, \nabla_\theta Q(s, a; \theta) \qquad (2)$$

Note that this is analogous to (1) up to an additional factor of $\Pr(s, a)$. This factor effectively scales the learning rate in proportion to how frequently the state-action pair occurs in the MDP under the policy $\mu$. Hence, even under these rather favorable conditions (an unchanging policy with infinite training samples), DQN suffers from *multiplicity bias* due to the uniform distribution. Significantly, this bias is not unique to DQN and affects other off-policy deep RL methods like DDPG [8], ACER [15], TD3 [5], and SAC [6].

## 3 STRATIFIED EXPERIENCE REPLAY

According to our theory, an ideal experience replay strategy would sample state-action pairs in inverse proportion to their relative frequencies under the stationary distribution. While it is not tractable to directly compute this distribution for high-dimensional environments, our agent has the advantage of a large replay memory at its disposal; hence, sample-based approximations are feasible.

Recall from Section 2 that the sampling probability under the uniform distribution factors: $\Pr(s, a, r, s') = \Pr(s' \mid s, a) \Pr(s, a)$. Dividing this by $\Pr(s, a)$ to eliminate the multiplicity bias, and then normalizing to make the probabilities sum to 1 over the set $\mathcal{S} \times \mathcal{A} \times \mathcal{S}$, we arrive at the ideal sampling distribution: $\Pr(s' \mid s, a) \,/\, |\mathcal{S} \times \mathcal{A}|$. Remarkably, this indicates that we can sample from two uniform distributions in succession to counter multiplicity bias. We call this *Stratified* Experience Replay[2] (SER) in which we first uniformly sample an antecedent state-action pair $(s, a)$ from $D$ and then uniformly sample a consequent reward-state pair $(r, s')$ from the transitions observed in $(s, a)$. By utilizing this two-step sampling strategy, we are able to achieve a reasonable approximation[3] to the ideal distribution without needing to explicitly compute these probabilities. Data Structure 1 outlines an efficient implementation of SER that avoids an expensive search over the replay memory and thereby maintains a sampling cost of $O(1)$.

[1] The reward $r = R(s, a, s')$ is deterministic and does not influence the probability.
[2] Our approach should not be confused with the recent method of the same name [13].
[3] It is not exact since, in practice, the replay memory will generally not contain $\mathcal{S} \times \mathcal{A}$ fully, nor will the experiences be collected from a single policy $\mu$. Future work that re-examines these simplifications could potentially improve empirical performance.

## 4 EXPERIMENTS

Code and implementation details for all experiments are online.[4] All networks were optimized using Adam [7]. In our first experiment, we trained a two-layer tanh DQN to solve Taxi [3] and FrozenLake [2], comparing the performance of SER against uniform experience replay. SER helps the agent learn significantly faster just by changing the sampling distribution (Figure 2).

Our second experiment compared the two sampling strategies when training a convolutional DQN on 11 Atari 2600 games within the ALE [1] following the procedures in [11] (excepting the use of Adam). While SER improved average performance in a majority of the games (Figure 3), the benefits were relatively modest compared to those of our first experiment. This is likely due to the high-dimensional nature of the games, wherein the majority of state-action pairs are visited no more than once.

Nevertheless, we were surprised to find that redundancy is still present in the games—particularly those where SER outperformed the baseline. For example, in Atlantis, we found that nearly 20% of the replay memory's samples were redundant after 1M training steps, and the most-visited sample was encountered over 250 times. We believe that SER's performance could be further improved by considering ways to count similar—not just identical—state-action pairs as being redundant (e.g. using density models [12]).

*Conclusion.* SER offers a theoretically well-motivated alternative to the uniform distribution for off-policy deep RL methods. By correcting for multiplicity bias, SER helps agents learn significantly faster in small MDPs, although the benefits are less pronounced in high-dimensional environments like Atari 2600 games. We see great promise in future methods that address scalability by exploring ways to generalize over similar state-action pairs during the stratification process.

## ACKNOWLEDGMENTS

This work was partially funded by US Army Research Office award W911NF-20-1-0265.

[4] https://github.com/brett-daley/stratified-experience-replay

## REFERENCES


[1] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. 2013. The Arcade Learning Environment: An Evaluation Platform for General Agents. *Journal of Artificial Intelligence Research* 47 (2013), 253–279.

[2] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. *arXiv:1606.01540* (2016).

[3] Thomas G Dietterich. 2000. Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition. *Journal of Artificial Intelligence Research* 13 (2000), 227–303.

[4] William Fedus, Prajit Ramachandran, Rishabh Agarwal, Yoshua Bengio, Hugo Larochelle, Mark Rowland, and Will Dabney. 2020. Revisiting Fundamentals of Experience Replay. *arXiv:2007.06700* (2020).

[5] Scott Fujimoto, Herke Van Hoof, and David Meger. 2018. Addressing Function Approximation Error in Actor-Critic Methods. *arXiv:1802.09477* (2018).

[6] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *arXiv:1801.01290* (2018).

[7] Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980* (2014).

[8] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous Control with Deep Reinforcement Learning. *arXiv:1509.02971* (2015).

[9] Long-Ji Lin. 1992. Self-Improving Reactive Agents Based on Reinforcement Learning, Planning and Teaching. *Machine Learning* 8, 3-4 (1992), 293–321.

[10] Ruishan Liu and James Zou. 2018. The Effects of Memory Replay in Reinforcement Learning. In *Allerton Conference on Communication, Control, and Computing*. IEEE, 478–485.

[11] 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. 2015. Human-Level Control through Deep Reinforcement Learning. *Nature* 518, 7540 (2015), 529–533.

[12] Georg Ostrovski, Marc G Bellemare, Aäron Oord, and Rémi Munos. 2017. Count-Based Exploration with Neural Density Models. In *International Conference on Machine Learning*. PMLR, 2721–2730.

[13] Anil Sharma, Mayank K Pal, Saket Anand, and Sanjit K Kaul. 2020. Stratified Sampling Based Experience Replay for Efficient Camera Selection Decisions. In *IEEE International Conference on Multimedia Big Data*. IEEE, 144–151.

[14] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement Learning: An Introduction*. MIT Press.

[15] Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. 2016. Sample Efficient Actor-Critic with Experience Replay. *arXiv:1611.01224* (2016).

[16] Christopher John Cornish Hellaby Watkins. 1989. *Learning from Delayed Rewards*. Ph.D. Dissertation. King's College.

[17] Shangtong Zhang and Richard S Sutton. 2017. A Deeper Look at Experience Replay. *arXiv:1712.01275* (2017).