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

Northeastern University

Harvard John A. Paulson School of Engineering and Applied Sciences

Brett Daley[1] Cameron Hickert[2] Christopher Amato[1]

[1]Northeastern University [2]Harvard University

Introduction

Deep Reinforcement Learning (Deep RL) often relies on off-policy Experience Replay [1] to decorrelate training data for neural networks [2]. Experiences are typically sampled from a uniform distribution over a replay memory.

We investigate to what extent the uniform distribution mitigates sample correlations.

Contributions

- We show that sampling from the uniform distribution causes *multiplicity bias* in Deep RL.
- Gradients are affected more by frequent experiences compared to rare ones.
- We propose an efficient stratified sampling scheme to cancel this effect.
- Our method improves learning speed in small environments.

Motivation

We can compare tabular Q-Learning [3] against Deep Q-Network (DQN) [2] to understand why the uniform distribution does not fully decorrelate data when using function approximation.

For both algorithms, suppose that the agent trains offline with the following assumptions:

1. The agent has an infinite-capacity replay memory D.
2. The agent executes a fixed policy μ for infinite time before training.

The probability that we sample an experience tuple (s, a, r, s') from D is theoretically $\Pr(s' \mid s, a) \cdot \Pr(s, a)$, which depends on the environment's transition function and the stationary distribution induced by μ, respectively.

Hence, we can compute the *expected* updates of these methods over all possible samples in D:

Q-Learning: $$Q(s,a) \leftarrow Q(s,a) + \alpha \cdot \sum_{s' \in \mathcal{S}} \Pr(s' \mid s, a) \cdot \delta(s, a, s')$$

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

Multiplicity bias (red term) arises for DQN, since gradient updates are not conditionally independent, unlike tabular updates. Effectively, the learning rate α is scaled by the relative frequency of the state-action pair (s,a), despite sampling from a uniform distribution on D.

Method

In theory, we could counteract multiplicity bias with importance sampling, but this is intractable. Instead, we can sample from two uniform distributions in succession:

1. Sample an antecedent state-action pair (s, a) from D.
2. Sample a consequent reward-state pair (r, s') from the transitions observed in (s, a).

This samples transitions in inverse proportion to their frequency of occurring. We call this strategy Stratified Experience Replay (SER) (Figure 1).

Stratified Experience Replay

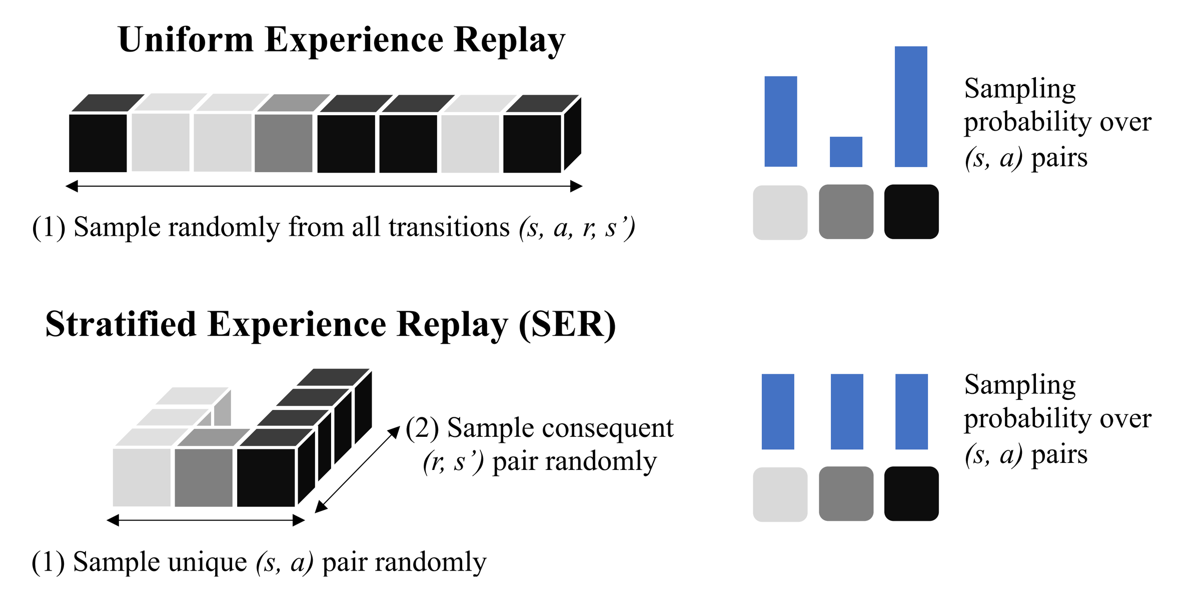

Figure 1: A graphical comparison of uniform (top) and stratified (bottom) sampling strategies.

Implementation

SER can be efficiently realized using a hash table and an array. This preserves $O(1)$ insertion, sampling, and deletion complexities.

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 $H[(s,a)] \leftarrow \textit{empty queue}$
  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

Experiments

We compared SER against the uniform distribution when training DQN in two experiments:

- Two-layer ReLU network on two gridworld environments (Figure 2).
- Five-layer convolutional network on eleven Atari 2600 games (Figure 3).

Code and implementation details for all experiments are available online:
`https://github.com/brett-daley/stratified-experience-replay`

Results

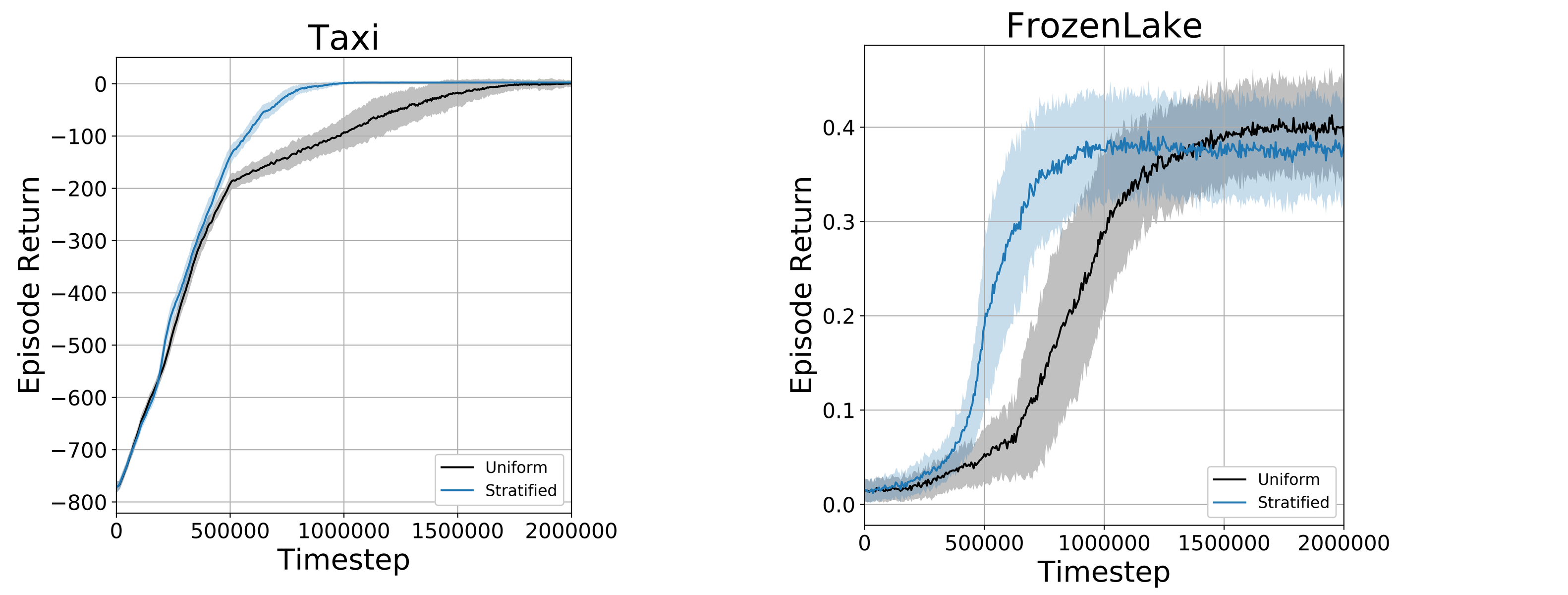

Figure 2: SER performance compared against a uniform baseline on two environments, averaged over 100 trials.

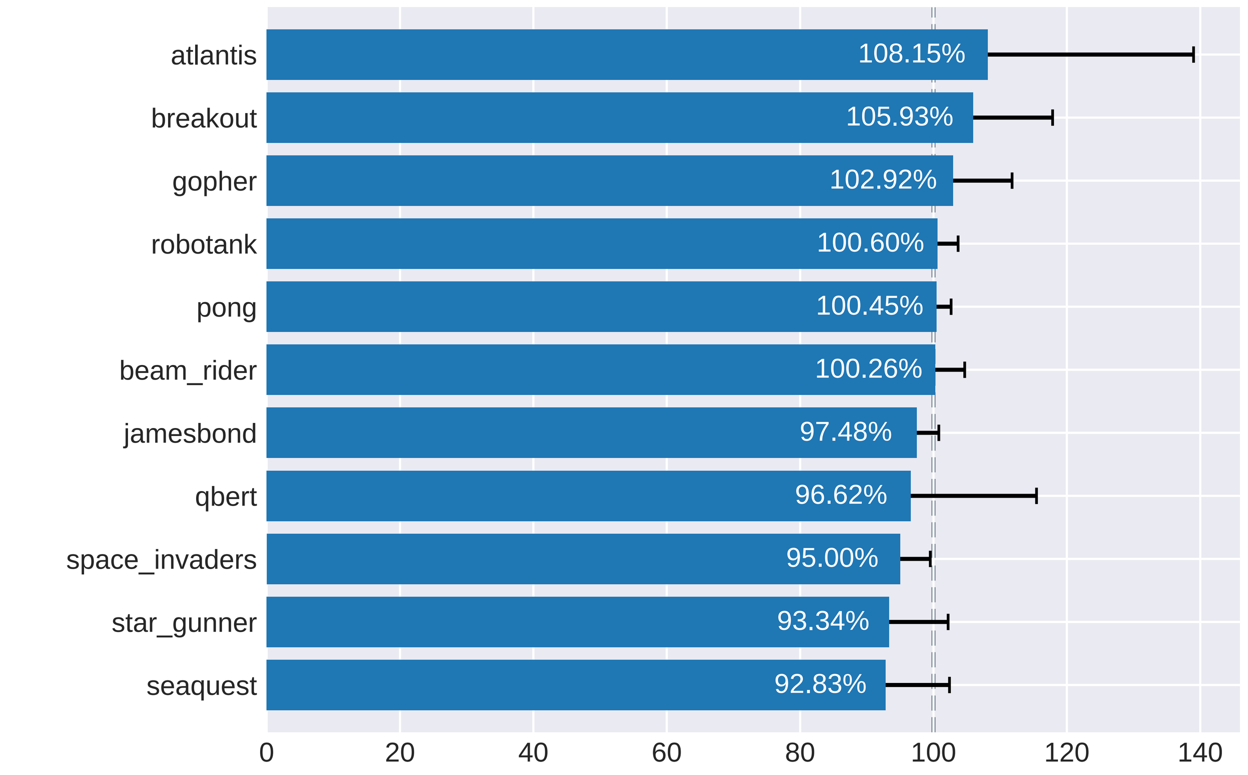

Figure 3: Average episode score of SER throughout training on eleven Atari games, relative to that of the uniform baseline, i.e. $100 \times$ (stratified − random) / (uniform − random). Results were averaged over 5 trials.

Conclusions

- Deep RL with Experience Replay is affected by multiplicity bias, even when sampling from the uniform distribution.
- Stratified Experience Replay (SER) uses two uniform distributions to counteract bias without needing to compute sampling probabilities.
- SER learns faster in small environments; scalability must be addressed in future work.

References

[1] Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8(3-4):293--321, 1992.

[2] 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.

[3] Christopher John Cornish Hellaby Watkins. *Learning from Delayed Rewards.* PhD thesis, King's College, 1989.