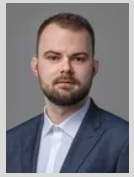
# Reliability-adjusted Prioritized Experience Replay

International Conference on Learning Representations 2026

**Leonard S. Pleiss**
Technical University of Munich
leonard.pleiss@tum.de

**Tobias Sutter**
University St.Gallen
tobias.sutter@unisg.ch

**Maximilian Schiffer**
Technical University of Munich
schiffer@tum.de

# Many offline reinforcement learning agents rely on *experience replay* – delayed learning on previously gathered environmental interactions – to improve their policy

## Experience Replay

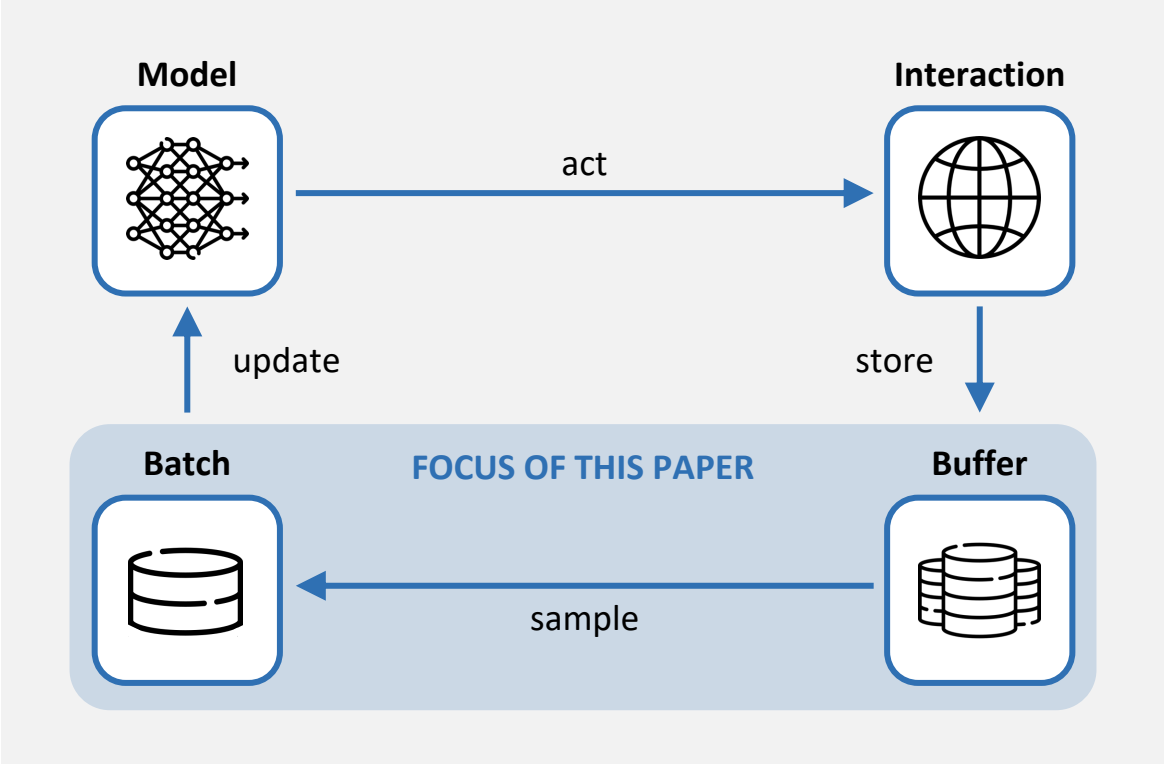



**Figure 1.** The experience replay mechanism.

**1992**

*Lin*[1]

**Experience replay**

- Traditionally: Uniform sampling from the Replay Buffer (each sample is replayed approximately equally often)

**2015**

*Schaul et al.*[2]

**Prioritized Experience Replay (PER)**

- Proposed that different samples vary in importance
- Temporal Difference (TD) error as proxy for importance
- Remains sole prioritization scheme used across state-of-the-art algorithms to this day[3, 4, 5, 6]

**2026**

*Pleiss et al.*

**Reliability-adjusted prioritized experience replay (ReaPER)**

- TD errors vary in reliability, which affects sample importance
- Given equal TD errors, samples with *more reliable* errors are more important

# Metaphorically speaking, TD errors indicate *surprise*, that is, they quantify the difference between what the agent *believed would happen* versus what *actually happened*

TD-Errors

TD error | Target Q-Value | Q-Value

$$\delta_t = \mathcal{R}_{t+1} + (1 - d_{t+1}) * \gamma * \max_a Q(S_{t+1}, a) - Q(S_t, A_t)$$

*TD error*

*Immediate reward*

*Expected return after performing action $A_t$ and observing the new state*

*Expected return when performing action $A_t$ in state $S_t$*

## Key takeaway

In temporal difference learning, the agent relies on its understanding of the subsequent state to improve their assessment of the current state. As such, the reliability of a temporal difference error relies on the agents‘ understanding of the subsequent state.

# The reliability of any transition‘s TD error depends on the TD errors of subsequent transitions within the same episode

## Intuition

*Example*

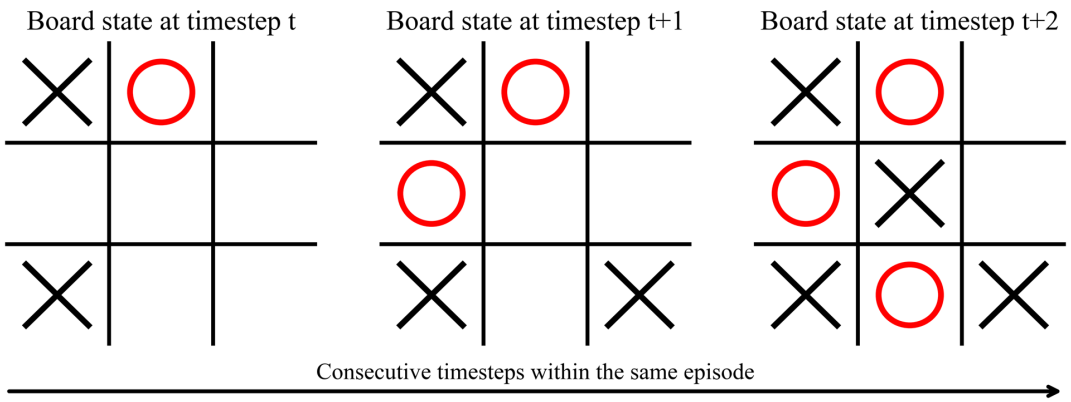



**Figure 2** displays subsequent states from a Tic Tac Toe game.

- All states are lost for the circles player under optimal play
- Recognizing that t+1 is a losing state is generally easier than recognizing t as such
- Once t+1 is identified as losing, identifying t as such becomes more straightforward

**Understanding of a state necessitates an understanding of subsequent states.**

**1**

### Unreliable TD errors can degrade learning

Low reliability TD errors may induce updates which lead to divergence, i.e., shifting the estimate away from its unknown true value.

**2**

### Terminal transitions yield reliable target errors

Target errors for terminal transitions are given by the environment and do not rely on model estimates. They are therefore perfectly reliable.

**3**

### Reliability propagates backwards

An update which improves the estimation accuracy of any given transition improves the target value reliability of its predecessor.

# We propose a novel reliability score for TD errors, which serves as the backbone for the ReaPER algorithm

ReaPER

## Reliability score

*Sum of TD errors of subsequent transitions in the same trajectory*

$$R_t = 1 - \frac{\sum_{i=t+1}^{n} \delta_i^+}{\sum_{i=1}^{n} \delta_i^+}$$

*Sum of TD errors of all transitions in the same trajectory*

## Importance criterion

*Reliability-adjustment*

$$\varphi_t = R_t * \delta_t^+$$

*Prioritized Experience Replay*

## Full ReaPER algorithm

We need to resolve 4 more challenges to obtain the final ReaPER algorithm:

1. *How do we efficiently update sample priorities?*
2. *How do we balance prioritization and coverage?*
3. *How do we address the i.i.d. violation?*
4. *How do we obtain reliabilities for unfinished episodes?*

*Solved mostly as in Prioritized Experience Replay*

*Solved via conservative proxy*

# We formally show that the reliability-based sampling accelerates convergence, and empirically validate our theoretical results

## Empirical & formal fndings

### Formal results

- We formally show that the proposed reliability score bounds the true target bias
- Based on this finding, we prove a convergence hierarchy of sampling strategies, ReaPER ≥ PER ≥ Uniform
- We further show that reliability-aware sampling reduces variance

*For details, we refer to the full paper*

### Empirical results

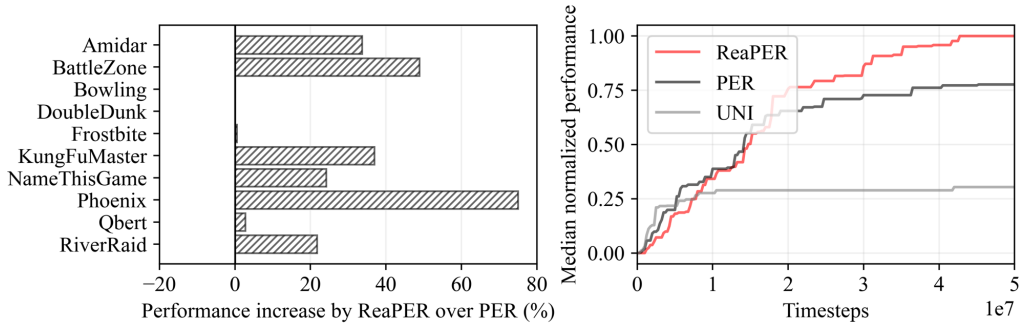



**Figure 3.** Peak performance on the Atari-10[7] benchmark.

ReaPER consistently outperforms both baselines

- 22.97% higher median peak score than PER
- 229.78% higher peak score than uniform replay

Under partial observability, ReaPER's edge over PER grows to 34.98%

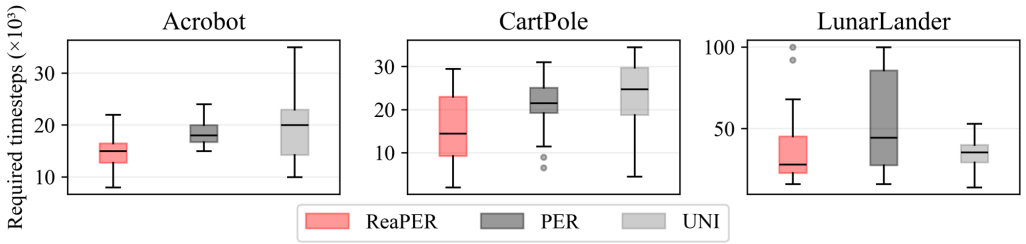



**Figure 4.** Timesteps to reach maximum performance in continuous control[8].

ReaPER consistently converges the fastest

- 25-41% faster than uniform sampling
- 17-33% faster than PER

# We briefly summarize our key contributions and limitations below. If you have further questions, feel free to get in touch!

## Discussion & further information

***Summary***

| | |
|---|---|
| **1** | We introduced the concept of TD-error reliability and established that it relies on the errors of subsequent states |
| **2** | Based on this insight, we proposed a novel reliability score for TD errors |
| **3** | We derived the ReaPER algorithm by extending the canonical PER framework with a reliability adjustment |
| **4** | We provided a formal analysis of the reliability metric, proving its soundness as a sample selection criterion |
| **5** | We empirically validated our findings, showing that ReaPER improves convergence speed and peak performance |

***Limitations***

| | |
|---|---|
| **A** | ReaPER relies on terminal states and is therefore not applicable in infinite horizon settings |
| **B** | ReaPER tracks cumulative sums over TD-errors to compute reliability, incurring some computational overhead |

***Further information***

**Paper**
https://iclr.cc/virtual/2026/poster/10008022

**Implementation**
https://github.com/leonardpleiss/ReaPER

**Contact**
leonard.pleiss@tum.de

# Key references


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

[2] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized Experience Replay. 2015

[3] 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, 2017.

[4] Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, and Charles Blundell. Agent57: Outperforming the Atari Human Benchmark, 2020a.

[5] Max Schwarzer, Johan Obando-Ceron, Aaron Courville, Marc Bellemare, Rishabh Agarwal, and Pablo Samuel Castro. Bigger, Better, Faster: Human-level Atari with human-level efficiency, 2023.

[6] Shengjie Wang, Shaohuai Liu, Weirui Ye, Jiacheng You, and Yang Gao. EfficientZero V2: Mastering Discrete and Continuous Control with Limited Data, 2024.

[7] Matthew Aitchison, Penny Sweetser, and Marcus Hutter. Atari-5: Distilling the Arcade Learning Environment down to Five Games, 2022.

[8] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. Gymnasium: A Standard Interface for Reinforcement Learning Environments, 2024.