Improving the Performance of Drone Navigation Training in Deep Reinforcement Learning Simulations

C. Zhou*, J. Cała†, S. McGough*

*School of Computing, †National Innovation Centre for Data, Newcastle University

Background

With the widespread adoption of Unmanned Aerial Vehicles (UAV)s across various sectors, their potential in road maintenance has become increasingly evident. To effectively identify common road quality issues such as potholes, faded markings and defects in signage, we endeavoured to use a simulated environment and train a deep reinforcement learning (DRL) agent which will enable UAVs recognise road defects. Using high-resolution cameras and advanced image processing techniques during ultra-low altitude flights UAVs can become indispensable tool to improve the quality of our roads.

Training Performance Bottleneck in Deep Reinforcement Learning

In our training process, we encountered significant performance bottlenecks and low-level GPU usage and, therefore, decided to delve into an analysis of the computational load across the entire training workflow. We assessed the impact of sampling and vectorization on the performance of the Prioritised Experience Replay (PER) buffer and conducted extensive tests using the Ray RL benchmarks within Atari environments.

Our findings revealed that as the batch size increased, the number of new transitions that could be acquired from the environment within the same time frame decreased. That was caused by excessive time consumption in Experience Replay operations. Additionally, we observed that the Ape-X framework exhibited similar challenges as PER. Thus, we propose to address these issues through Block Experience Replay, our new method to manage experience replay buffer.

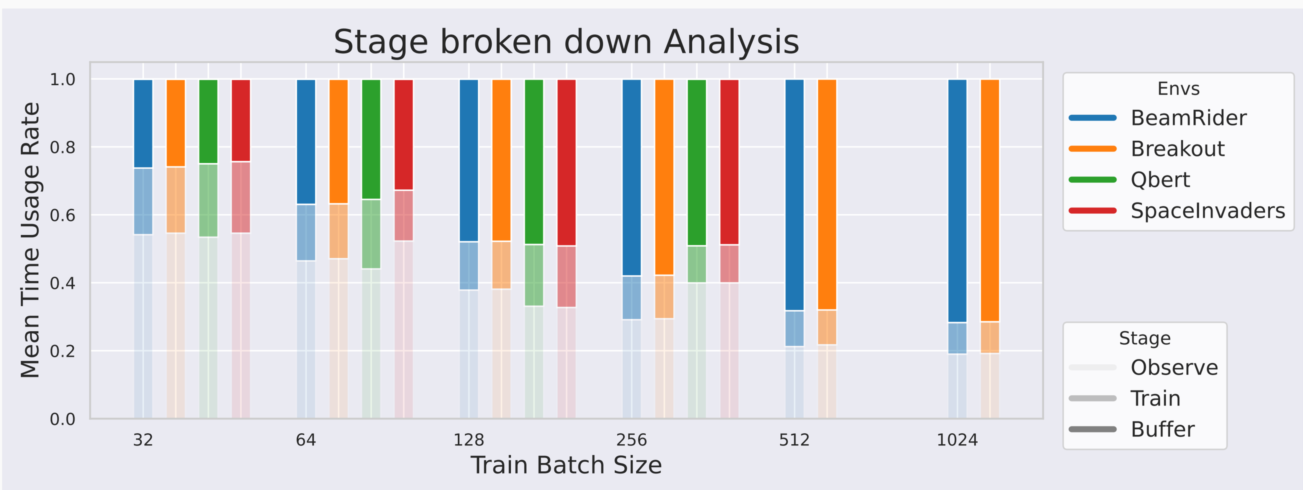

Fig. 1: Time used in training.

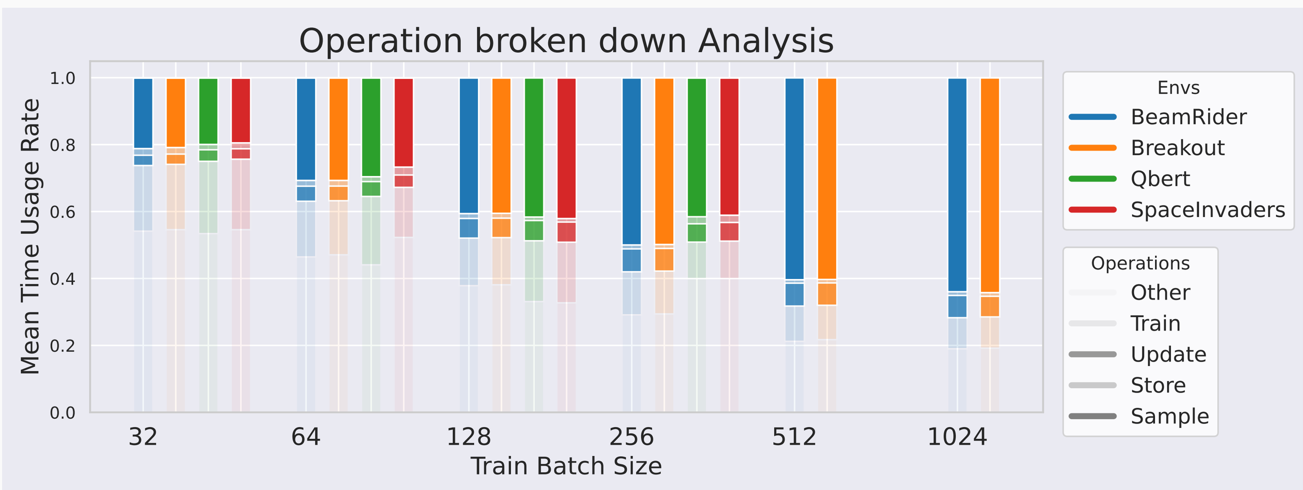

Fig. 2: Time used in experience replay operations.

Block Experience Replay

We designed and implemented a new general structure of ER which joins benefits of PER, HER, and Ape-X, and shows a significant improvement in reducing the cost in experience replay operation.

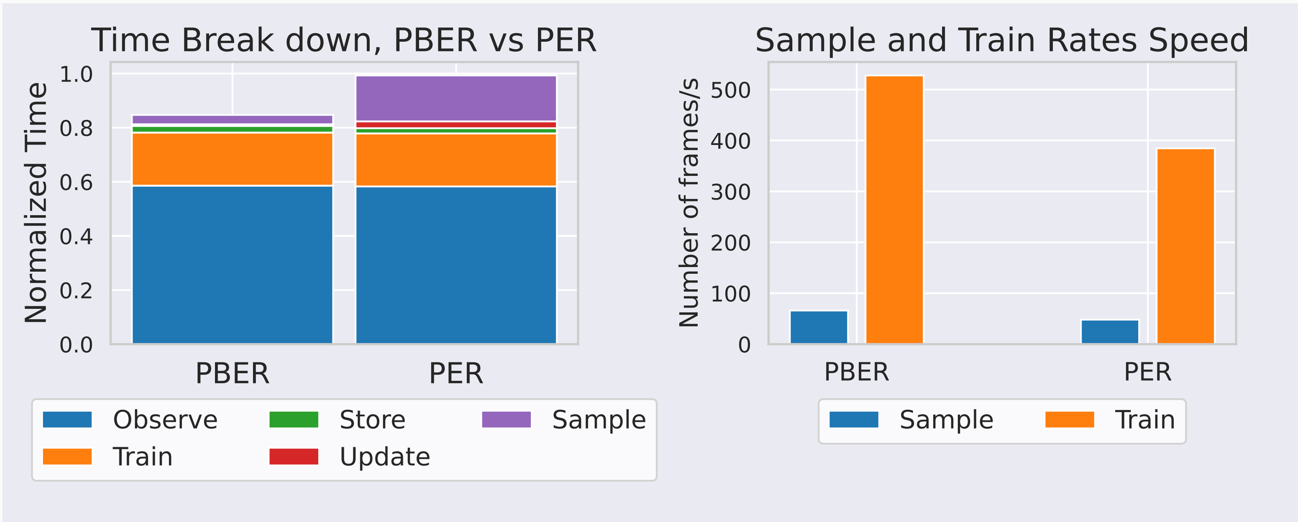

Fig. 3: Time usage difference in PBER vs PER

Experiments on JADE

We evaluated our method in the Atari Games environments by analysing the number of transitions processed. In Ape-X algorithm, we set the number of `Actors` to 4 to optimise performance on `JADE Small` and ensure stable results. Uniform hardware requirements were set for all experiments.

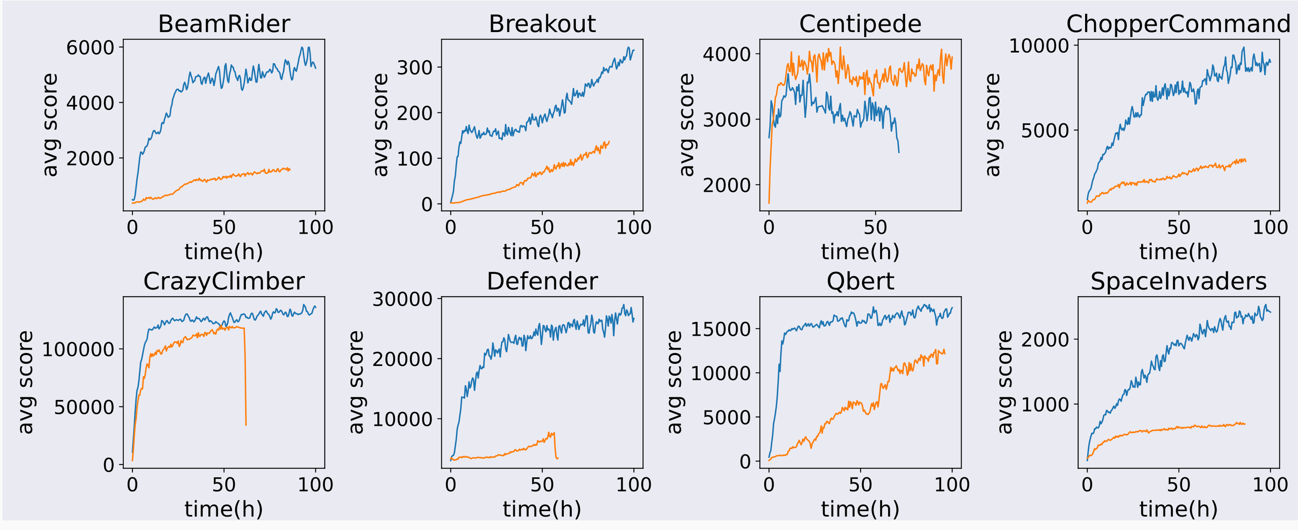

Fig. 4: Example learning curves in DPBER vs DPER.

Compared to Ape-X DDQN with DPER (orange lines), the Ape-X DDQN with DPBER (blue lines) trained a notably larger number of frames and consistently outperformed in most tests.

We tested BER across 24 Atari game environments, comparing ER vs BER, PER vs PBER, and DPBER vs DPER. It resulted in 164 runs, each lasting 100 hours. On a single local system this would equate to 2 years of run time. However, on JADE we were able to complete all experiments in just 17 days.

National Innovation Centre Data
Powered by Newcastle University

National Innovation Centre for Data aims to strengthen UK economy by helping public and private sector organisations learn new data skills and become more productive. Find out more at: `https://nicd.org.uk`.

LaTeX TikZposter