PERFORMANCE COMPARISON OF VALUE-BASED AND POLICY-BASED DEEP REINFORCEMENT LEARNING METHODS ON ATARI BREAKOUT

Ria Manu Joseph

IIT(ISM)Dhanbad
Computer Science and Engineering

ABSTRACT

This project explores the application of deep reinforcement learning (RL) techniques to train an agent that achieves expert-level performance in the Atari game Breakout. Using the Gymnasium interface with the Arcade Learning Environment (ALE), we implement and evaluate 3 standard deep RL algorithms—specifically vanilla Deep Q-Networks (DQN), Proximal Policy Optimization (PPO) and Advantage Actor Critic (A2C) —to determine their effectiveness in learning from high-dimensional pixel inputs. Each agent is trained using frame-stacked grayscale observations processed by convolutional neural networks. Performance is measured using average and maximum game scores. This study aims to highlight the relative strengths of value-based methods and a comparison with a policy-based method in discrete-action visual environments (such as Breakout) and provides insights into algorithmic trade-offs for pixel-based deep RL tasks.

1. INTRODUCTION

Reinforcement Learning (RL) has emerged as a powerful tool for training agents to make sequential decisions through interaction with an environment. With the help of Deep Reinforcement Learning (DRL), agents can learn directly from high-dimensional sensory inputs such as images, enabling breakthroughs in domains like game play, robotics, and autonomous control. Atari games, in particular, have served as a key benchmark for evaluating the performance of DRL algorithms due to their well-defined environments and clear reward structures. Among Atari environments, Breakout provides a suitable field, due to its simple yet challenging dynamics, requiring both precise control and strategic planning to maximize score. The agent must learn to track the ball, control the paddle, and adapt its strategy as the brick configuration changes, making it an effective environment for evaluating both value-based and policy-based algorithms. The choice of focus for this study is on three prominent DRL algorithms: Deep Q-Networks (DQN), a value-based method that approximates the optimal Q-function using deep neural networks; Proximal Policy Optimization (PPO), a policy-gradient approach known for stable and efficient updates; and Advantage Actor-Critic (A2C), a synchronous version of the A3C algorithm [1] that combines policy and value estimation. Together, these algorithms represent distinct groups within DRL—value-based, policy-based, and actor-critic methods—allowing for a comprehensive and well-rounded comparison. The objective of this research is to evaluate and compare the performance of DQN, PPO, and A2C in training an agent to play Atari Breakout. We aim to analyze their learning efficiency, stability, and achieved game performance under similar experimental conditions. The agents are trained on pixel-level observations processed via convolutional neural networks (CNNs), with preprocessing techniques such as frame skipping, grayscale conversion, and frame stacking to enhance learning stability. Specifically, we examine (1) which algorithms are most effective for learning from high-dimensional visual inputs, (2) how preprocessing techniques impact stability and convergence, and (3) how network architecture affects the agent's ability to discover complex strategies. Through this investigation, we aim to provide empirical insights into the trade-offs between different DRL algorithms when applied to visually rich, discrete-action environments.

2. BACKGROUND AND RELATED WORK

The field of Reinforcement Learning (RL) is grounded in the idea of training agents to make sequential decisions through trial and error, guided by rewards. Early RL approaches such as Q-learning and SARSA established foundational principles for value-based learning in discrete environments. However, their scalability was limited when applied to high-dimensional state spaces, as traditional tabular methods could not efficiently represent continuous or visual input domains. The emergence of Deep Reinforcement Learning (DRL) overcame this limitation by integrating deep neural networks as function approximators. A landmark advancement occurred with Deep Q-Network (DQN), introduced by Minh et al. [2], which combined Q-learning with convolutional neural networks (CNNs) to learn directly from raw pixel data. Using experience replay and a target network for stability, DQN achieved human-level performance across multiple Atari 2600 games, establishing a new benchmark for DRL research. Subsequent works extended and improved upon

DQN. Variants such as Double DQN (van Hasselt et al., 2016), Dueling DQN (Wang et al., 2016), and Rainbow DQN (Hessel et al., 2018) incorporated techniques like double estimation, advantage decomposition, and prioritized experience replay to enhance learning stability and performance. These algorithms collectively refined the efficiency and robustness of value-based methods in visually rich environments.

In parallel, policy-gradient and actor-critic methods emerged as an alternative paradigm. Instead of estimating action values, these methods directly optimize a parameterized policy. The Advantage Actor-Critic (A3C) algorithm (Mnih et al., 2016) introduced asynchronous training to improve sample efficiency and reduce variance. Its synchronous variant, A2C, simplified implementation while maintaining competitive performance. Proximal Policy Optimization (PPO), proposed by Schulman et al. (2017) [3], further advanced policy optimization by introducing a clipped objective function, providing a balance between learning stability and policy improvement.

Research on Atari Breakout has continued to serve as a crucial benchmark for evaluating DRL methods due to its structured reward system, moderate difficulty, and potential for emergent strategy formation. Prior studies have demonstrated that algorithms like DQN can discover advanced strategies such as tunneling through the brick wall, while PPO and A2C tend to exhibit smoother and more stable training curves under comparable conditions. Recent efforts have also explored enhancements such as curiosity-driven exploration, reward shaping, and improved preprocessing pipelines to accelerate convergence and improve generalization across Atari games.

Overall, these prior works establish a rich foundation for this study. By comparing DQN, PPO, and A2C under a unified experimental framework in the Breakout environment, this research aims to contribute empirical insights into how algorithmic design choices and preprocessing techniques influence learning efficiency, stability, and final gameplay performance.

3. IMPLEMENTATION

The experiments were conducted using the Farama Gymnasium interface [4] and the Arcade Learning Environment (ALE) [5] to provide access to the Breakout-v5 environment. Three deep reinforcement learning algorithms—Deep Q-Network (DQN), Proximal Policy Optimization (PPO), and Advantage Actor-Critic (A2C)—were implemented using Stable-Baselines3 [6] in Python. This framework offers standardized, reproducible implementations of state-of-the-art RL algorithms, ensuring a consistent basis for comparison.

To improve sample efficiency and stabilize learning, the environment was wrapped with two key preprocessing components: AtariPreprocessing and FrameStacking, provided by Gymnasium's wrapper suite. AtariPreprocessing converts the raw RGB frames to grayscale, resizes them to 84×84 pixels, and applies frame skipping (typically every 4 frames), effectively reducing temporal redundancy and computational cost. The FrameStack wrapper then combines the most recent four frames into a single observation tensor, enabling the agent to infer motion and temporal context—an essential feature for environments with partial observability such as Breakout.

For each algorithm, custom scripts were developed to manage the full training and evaluation pipeline. Progress was continuously tracked throughout training, with metrics such as episodic reward, loss, and timestep count logged for visualization. The training statistics were plotted using Matplotlib, allowing for an interpretive comparison of convergence rates and performance stability across algorithms.

The choice of hyperparameters followed the default configurations recommended in the Stable-Baselines3 documentation, with selective adjustments made to suit the Breakout task. In particular, techniques such as linear decay of the learning rate and clip range (for PPO) were employed to enhance stability and prevent overfitting during policy updates. The convolutional neural network (CNN) architecture adopted the standard structure for image-based RL tasks: stacked 84×84 frames passed through convolutional layers followed by fully connected layers that output either Q-values (for DQN) or policy and value estimates (for PPO and A2C).

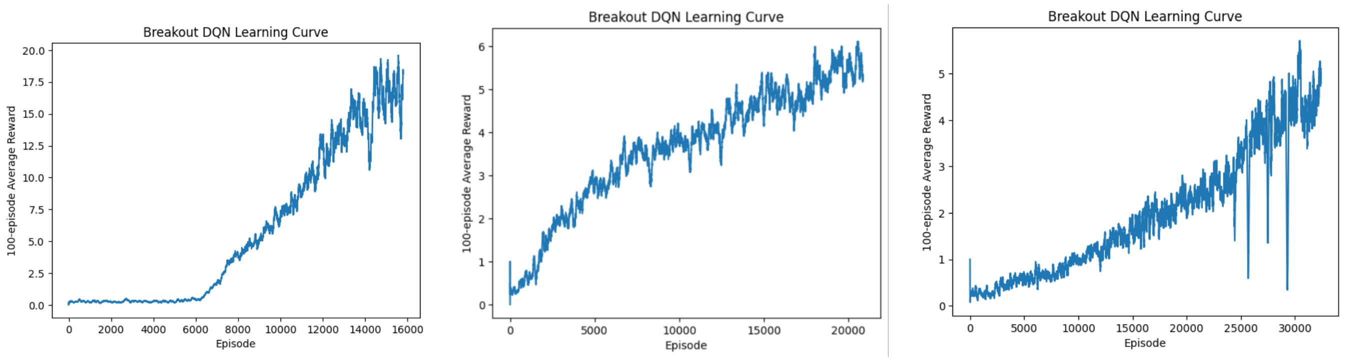
Each agent was trained for three million timesteps, which, while considerably smaller than large-scale experiments, was sufficient to observe meaningful learning progress and performance differentiation among the algorithms. For comparison, the original DeepMind DQN paper (Mnih et al., 2013) trained for approximately 50 million frames to achieve superhuman performance in Atari games such as Breakout, whereas modern replications typically require around 10 million frames to exhibit advanced behaviors like tunneling through the brick wall. In contrast, our study focuses on understanding relative learning dynamics, stability, and early-stage strategy formation within a controlled computational budget.

4. RESULTS

4.1. Quantitative Performance

The average episodic reward over the course of training was used as the primary indicator of each algorithm's performance. The figures below show the average reward (smoothed over the last 100 episodes) versus episode for DQN, PPO, and A2C.

From the training curves, DQN starts with minor increases and then increases sharply. It achieved the highest average reward. PPO demonstrates gradual increases throughout training with low variance. PPO is more stable and consistent when compared to the other two. Both PPO and A2C experience stronger fluctuations. A2C appears to be lower-performing.



(a) DQN

(b) PPO

(c) A2C

Fig. 1: Average reward over 100-episode windows for DQN, PPO, and A2C.

At the end of training (three million timesteps), the final evaluation rewards were approximately: PPO: 7.0, DQN: 9.0, A2C: 6.0 These values are just for demonstration of the training results.

4.2. Gameplay Observations

[For video viewing, please refer to the README file of the associated GitHub repository][1] The DQN model tends to keep the paddle at the ends. Perhaps, so as to discover tunneling behaviour, by breaking through the sides. The PPO model has the paddle moving unnecessarily. The A2C model shows good control of the paddle, but tended to lose precision at higher ball speeds, suggesting underfitting to the environment's increasing difficulty.

5. DISCUSSION AND CONCLUSION

From the graphs and videos, we make the following inferences : DQN eventually starts learning meaningful strategies but perhaps requires more training. The fluctuations in the graph can be a result of sensitivity to Q-value bootstrapping and replay buffer sampling. Due to clipping, PPO has a smoother graph as result of prevention of large, drastic updates. A2C lacks clipping so it may lead to more drastic updates; A2C is also known to be more sensitive to hyperparameters, as demonstrated in the Comparative Study Paper (Delafuente, Guerra) [7]. Frame stacking and preprocessing likely helped all agents learn faster, especially PPO and A2C, as their smooth curves suggest better temporal understanding. The network architecture—particularly the depth and configuration of convolutional layers—determines how the agent processes visual information from the environment. If the CNN is too shallow, it struggles to detect higher-level spatial relationships (like the ball's trajectory or tunnel formation). If it's too deep, it may overfit or train too slowly due to vanishing gradients and excessive parameter count. We use the standard settings as provided by Stable-Baselines3. For details such as layer structure and default hyperparameters, refer to the official documentation [2].

Due to time constraints and limited access to high-performance GPUs, the models were trained for fewer frames than would be required to achieve expert-level performance. Consequently, the results presented reflect only the early-to-mid stages of learning. Extending training over more frames would likely lead to improved performance and more stable learning curves. We have only seen the surface level of what these algorithms are truly capable of.

[1] GitHub Repository

[2] Stable-Baselines3 Documentation

6. REFERENCES

[1] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," *International Conference on Machine Learning (ICML)*, 2016.

[2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[3] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[4] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al., "Gymnasium: A standard interface for reinforcement learning environments," *arXiv preprint arXiv:2407.17032*, 2024.

[5] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The arcade learning environment: An evaluation platform for general agents," *Journal of Artificial Intelligence Research*, vol. 47, pp. 253–279, jun 2013.

[6] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.

[7] Neil de la Fuente and Daniel A. Vidal Guerra, "A comparative study of deep reinforcement learning models: Dqn vs ppo vs a2c," *arXiv preprint arXiv:2407.14151*, 2024.