Comparative Study of Deep Reinforcement Learning Methods on the Game of Snake

Fotios Kapotos   Adonis Jamal   Jean-Vincent Martini

CentraleSupélec

1. Motivation and Problem Definition

Reinforcement Learning (RL) has demonstrated remarkable success in sequential decision-making problems such as Atari games [2], Go [5], and robotics control [6].

The Snake game presents a compelling environment for studying Reinforcement Learning (RL) due to its simple rules yet complex strategic requirements. Unlike many RL benchmarks, Snake requires agents to balance immediate rewards (food collection) with long-term survival (avoiding self-collision), making it an ideal testbed for exploring how different RL algorithms handle credit assignment and exploration-exploitation tradeoffs.

The game's dynamics create several challenging scenarios: as the snake grows, the state space complexity increases, navigation becomes progressively constrained, and the agent must plan increasingly longer trajectories to avoid trapping itself. These characteristics mirror real-world sequential decision-making problems where available actions become limited over time and where greedy strategies lead to failure.

Our primary objective is to conduct a systematic comparison of modern deep RL algorithms on the Snake environment, investigating how algorithmic choices, reward shaping strategies, and state representations affect learning efficiency and final performance.

2. Related Work

Deep Q-Networks (DQN) introduced by [2] demonstrated that deep neural networks can approximate action-value functions directly from high-dimensional inputs. Extensions such as Double DQN [7] improve stability by reducing overestimation bias.

Policy-gradient methods such as Proximal Policy Optimization (PPO) [4] and Advantage Actor-Critic (A2C) [3] provide alternative optimization strategies with improved convergence properties and robustness.

Game-based RL benchmarks have historically focused on Atari environments using the Arcade Learning Environment. However, smaller grid-based environments like Snake provide a controlled setting for studying reward shaping, representation learning, and exploration strategies.

Tree-search methods such as Monte Carlo Tree Search (MCTS) [1] have shown effectiveness in planning-based settings. We will optionally explore hybrid approaches combining learned policies with search-based planning.

3. Methodology

3.1. Environment Setup

We implement the Snake game as a Gymnasium-compatible environment, allowing for standardized interaction with RL algorithms. Thanks to the implementation of our custom wapper, we can easily modify the state representation and reward structure to test different configurations.

The problem is formulated as a Markov Decision Process (MDP) defined by:

State space: The environment will be represented under multiple state encodings. Examples include:
- Full grid representation (raw board matrix).
- Local egocentric view centered on the snake's head.
- Hand-crafted feature vector (distances to walls, food direction, collision indicators).

Action space: Discrete actions: {move forward, turn left, turn right}.

Reward function: We will compare different reward shaping strategies such as:
- Sparse rewards (food consumption + terminal penalty).
- Dense rewards (distance-based shaping, survival reward, loop penalties).

3.2. Agents

We will implement and compare:
- Deep Q-Network (DQN) [2]: Value-based method with experience replay and target networks.
- Double DQN [7]: Extension of DQN that decouples action selection from evaluation to reduce overestimation bias.
- Advantage Actor-Critic (A2C) [3]: On-policy algorithm combining policy gradient with value function baseline to

reduce variance.

- Proximal Policy Optimization (PPO) [4]: Policy gradient method with clipped surrogate objective ensuring stable updates through trust region constraints.
- Random Agent: Baseline that selects actions uniformly at random to assess the difficulty of the environment and the effectiveness of learning.

All neural agents will share comparable architectures to ensure fair comparison. Hyperparameters will be tuned systematically.

4. Evaluation

Evaluation Protocol: We will train each agent for a fixed number of episodes (e.g., 10,000) and evaluate performance using multiple random seeds to assess learning stability and generalization.

Evaluation Metrics: We will focus on:
- Average score (food collected)
- Maximum length achieved
- Survival time (number of steps before termination)
- Reward value
- Sample efficiency (learning curve analysis)

Qualitative Analysis: We will analyze learned behaviors through visualizations of agent trajectories, action distributions, and state visitation patterns to understand how different algorithms navigate the tradeoffs inherent in the Snake environment. Results will be presented in the form of learning curves, box plots of final performance, and trajectory visualizations.

Comparaison with Tree Search: Eventually, we will use the learned policies to explore trees in order to evaluate the potential application of RL agents to new environments. This allows us to compare the performance to more traditional methods such as Depth-First Search (DFS) and Breadth-First Search (BFS) in the same environment, providing insight into the strengths and weaknesses of learned policies versus search-based approaches.

5. Expected Contributions

This project aims to provide a structured empirical study of how algorithmic design choices impact performance in a constrained, progressively complex RL environment.

First, we expect to provide a clear comparative analysis of value-based and policy-based methods on a growing-horizon task by evaluating DQN, Double DQN, A2C, and PPO in the Snake environment.

Second, we expect to contribute an in-depth study of reward shaping in a sparse-reward environment. Snake naturally induces delayed consequences (self-trapping occurs long after a greedy decision), making it a meaningful case study for understanding how dense shaping signals affect exploration and long-term planning.

Third, by comparing multiple state representations (raw grid, local view, and hand-crafted features), we aim to provide insight into representation learning in grid-based environments.

Finally, the project will result in a modular Gymnasium-compatible Snake environment and evaluation framework that supports systematic experimentation. This implementation can serve as a lightweight benchmark for studying algorithmic behavior in constrained navigation and survival-based RL tasks.

References

[1] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012. 1

[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. 1

[3] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PmLR, 2016. 1

[4] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 1, 2

[5] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529 (7587):484–489, 2016. 1

[6] Bharat Singh, Rajesh Kumar, and Vinay Pratap Singh. Reinforcement learning in robotic applications: a comprehensive survey. *Artificial intelligence review*, 55(2):945–990, 2022. 1

[7] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, 2016. 1