

# Feature Engineering Exploration in Mario Game AI

## Project Proposal

### Group Members:

Yushan Liu 2024214103  
Xintao Chao 2024213928  
Wenxuan Zhu 2024213949

November 27, 2024

The GitHub repository for this project is available at:  
[https://github.com/scpsyl/LFD\\_Final\\_Mario](https://github.com/scpsyl/LFD_Final_Mario).

# Introduction

This project explores feature engineering techniques in training an intelligent Mario game agent using reinforcement learning (RL). The goal is to analyze how modifications to observation space, action space, and reward space affect the agent’s learning efficiency and performance in the Super Mario Bros game environment.

## Background

The project is built upon the DI-adventure framework, which is designed for RL tasks in the Mario game environment. The agent learns to navigate and complete levels using Deep Q-Networks (DQN). The environment is ideal for experimentation due to its dynamic nature, high-dimensional observation space, and complex action requirements.

## Scope and Objectives

The main objective is to investigate how different feature engineering strategies impact the agent’s learning process. The specific goals include:

- Modifying observation space through techniques like image downsampling, frame stacking, and content simplification.
- Experimenting with action space configurations such as simplifying or enriching action sets and incorporating sticky actions.
- Exploring reward space modifications, such as sparse rewards and adding specific incentives (e.g., coin collection).

The ultimate goal is to find the most effective feature engineering strategies that enhance learning efficiency and performance while considering computational constraints.

## Proposed Methodology

### Baseline Setup

- Set up the DI-adventure framework with default observation, action, and reward spaces.
- Train the baseline model with one random seed and evaluate its performance in terms of rewards and gameplay efficiency.

### Feature Engineering Experiments

- **Observation Space:**
  - Implement frame stacking ( $\times 4$ ) to capture motion dynamics.
  - Simplify game content by switching to different environment versions (e.g.,  $v_1$ ,  $v_2$ ).

- Downsample image resolution to reduce computational load.
- **Action Space:**
  - Simplify actions (e.g., only right movement and jump: ['right'], ['right', 'A']).
  - Experiment with complex action sets (COMPLEX\_MOVEMENT) for richer gameplay.
  - Add sticky actions for increased randomness.
- **Reward Space:**
  - Implement sparse rewards to encourage goal-oriented behavior.
  - Introduce coin-based rewards to incentivize specific actions.

## Performance Evaluation

- Use TensorBoard to track reward trends, steps per episode, and Q-value dynamics.
- Evaluate performance using gameplay videos and Class Activation Mapping (CAM).
- Quantify key metrics: average reward, steps per episode, and learning curves.

## Environment and Tools

- **Platform:** Python 3.8
- **Libraries:** PyTorch 1.10.0, OpenCV, DI-engine
- **Environment:** Super Mario Bros RL simulation (DI-adventure framework)
- **Hardware:** GPU-enabled systems for efficient training

## Expected Outcomes

This project is expected to provide insights into how feature engineering impacts RL agent performance in dynamic environments. We aim to identify the most effective feature engineering strategies that improve learning efficiency, especially under computational constraints.

## Deliverables

- A trained Mario game agent with baseline and modified feature settings.
- A comprehensive analysis of experimental results.
- Project documentation and a presentation summarizing findings and conclusions.

This project aims to contribute valuable insights into the application of feature engineering techniques in RL tasks, particularly in dynamic video game environments like Super Mario Bros.

## References

- [1] OpenDILab DI-adventure. *GitHub Repository for RL Projects*. Available at: <https://github.com/opensailab/DI-adventure>.
- [2] OpenDILab DI-engine. *GitHub Repository for Reinforcement Learning Algorithms*. Available at: <https://github.com/opensailab/DI-engine>.
- [3] Gym Super Mario Bros. *Python API for Super Mario Bros Reinforcement Learning Environment*. Available at: <https://github.com/Kautenja/gym-super-mario-bros>.
- [4] Mnih, V., et al. *Human-level Control Through Deep Reinforcement Learning*. Nature, 2015. Available at: <https://www.nature.com/articles/nature14236>.
- [5] Dulac-Arnold, G., et al. *Action Space Shaping in Deep Reinforcement Learning*. Available at: <https://arxiv.org/pdf/2004.00980.pdf>.
- [6] Weng, L. *Reinforcement Learning Overview*. Available at: <https://lilianweng.github.io/posts/2018-02-19-rl-overview/>.