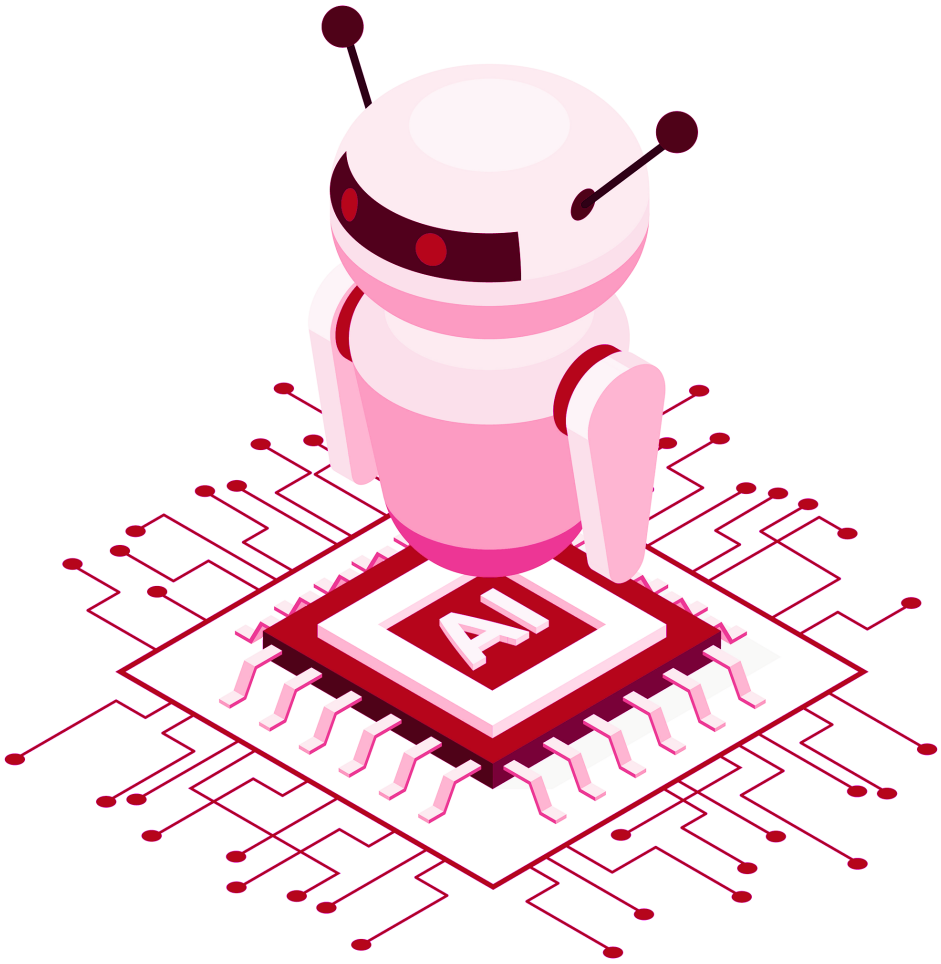

Deep Reinforcement Learning

Professor Mohammad Hossein Rohban

Homework 10:

Exploration in Deep Reinforcement Learning

By:

Amir Kooshan Fattah Hesari

401102191

RIML

Deep RL Course

Spring 2025

Contents

Grading

The grading will be based on the following criteria, with a total of 290 points:

Task	Points
Task 1: Bootstrap DQN Variants	100
Task 2: Random Network Distillation (RND)	100
Clarity and Quality of Code	5
Clarity and Quality of Report	5
Bonus 1	80

1 Task 1: Bootstrap DQN Variants

- The complete guidelines for implementing the Bootstrap DQN algorithm, including the RPF and BIV variants, are provided in the Jupyter notebook. You will find detailed instructions on how to set up the environment, implement the algorithms, and evaluate their performance.
- Make sure to read Guidelines section in the notebook carefully.

2 Task 2: Random Network Distillation (RND)

- You will implement the missing core components of Random Network Distillation (RND) combined with a Proximal Policy Optimization (PPO) agent inside the MiniGrid environment.
- **TODO:** You must complete the following parts:

File	TODO Description
`Core/model.py`	Implement the architecture of `TargetModel` and `PredictorModel`.
`Core/model.py`	Implement `_init_weights()` method for proper initialization.
`Core/ppo_rnd_agent.py`	Implement `calculate_int_rewards()` to compute intrinsic rewards.
`Core/ppo_rnd_agent.py`	Implement `calculate_rnd_loss()` to compute predictor training loss.

Table 1: Summary of required TODO implementations

- Questions:
  1. What is the intuition behind Random Network Distillation (RND)? Why does a prediction error signal encourage better exploration?
     Answer : RND works by creating an intrinsic motivation signal based on novelty. The less the predictor knows about a state, the more valuable it becomes to visit that state again. This prediction error functions as a curiosity-driven reward, helping agents discover useful behavior in otherwise reward-sparse environments.
  2. Why is it beneficial to use both intrinsic and extrinsic returns in the PPO loss function?
     Answer : Increasing the predictor proportion—which corresponds to masking a greater number of input features in the Random Network Distillation (RND) loss has been found to negatively impact the learning process. When more features are masked, the predictor network receives less information about the input state, making it significantly more challenging to accurately approximate the output of the fixed target network. As a result, the predictor generates higher prediction errors, not necessarily because the state is novel or informative, but simply due to the lack of input context. This leads to noisy and unreliable intrinsic rewards, which can confuse the agent and encourage it to explore in directions that are not actually useful or novel. In essence, excessive masking dilutes the quality of the intrinsic motivation signal, causing the agent to pursue misleading exploration strategies and ultimately slowing down or degrading the overall learning performance.
  3. What happens when you increase the `predictor_proportion` (i.e., the proportion of masked features used in the RND loss)? Does it help or hurt learning?
  4. Try training with `int_adv_coeff=0` (removing intrinsic motivation). How does the agent's behavior and reward change?

Setting int adv coeff = 0, which effectively disables the influence of intrinsic motivation, typically results in significantly slower and less effective exploration—particularly in environments with sparse or deceptive rewards. In the absence of an intrinsic reward signal, the agent has no internal drive to seek out novel states or actions, relying solely on external rewards that may be infrequent or delayed. This limitation becomes especially problematic in challenging environments like MiniGrid, where discovering complex reward pathways often requires persistent and strategic exploration. Without intrinsic motivation, the agent is far less likely to stumble upon these rewarding trajectories, leading to a failure to uncover critical states and ultimately causing it to converge to suboptimal policies.

When int adv coeff is set to zero, the intrinsic reward sharply diminishes—reflecting the absence of curiosity-driven learning. Meanwhile, the extrinsic reward remains relatively unchanged, though it exhibits slight fluctuations due to environmental noise. This further illustrates that without intrinsic incentives to explore, the agent's learning progress becomes limited, and its capacity to improve through novel experience is greatly reduced.

5. Inspect the TensorBoard logs. During successful runs, how do intrinsic rewards evolve over time? Are they higher in early training?

   Answer :
   - Initially increases as the target network explores the environment and builds a foundational understanding of various states.
   - Gradually decreases over time as the states become more familiar and the predictor network's errors diminish.

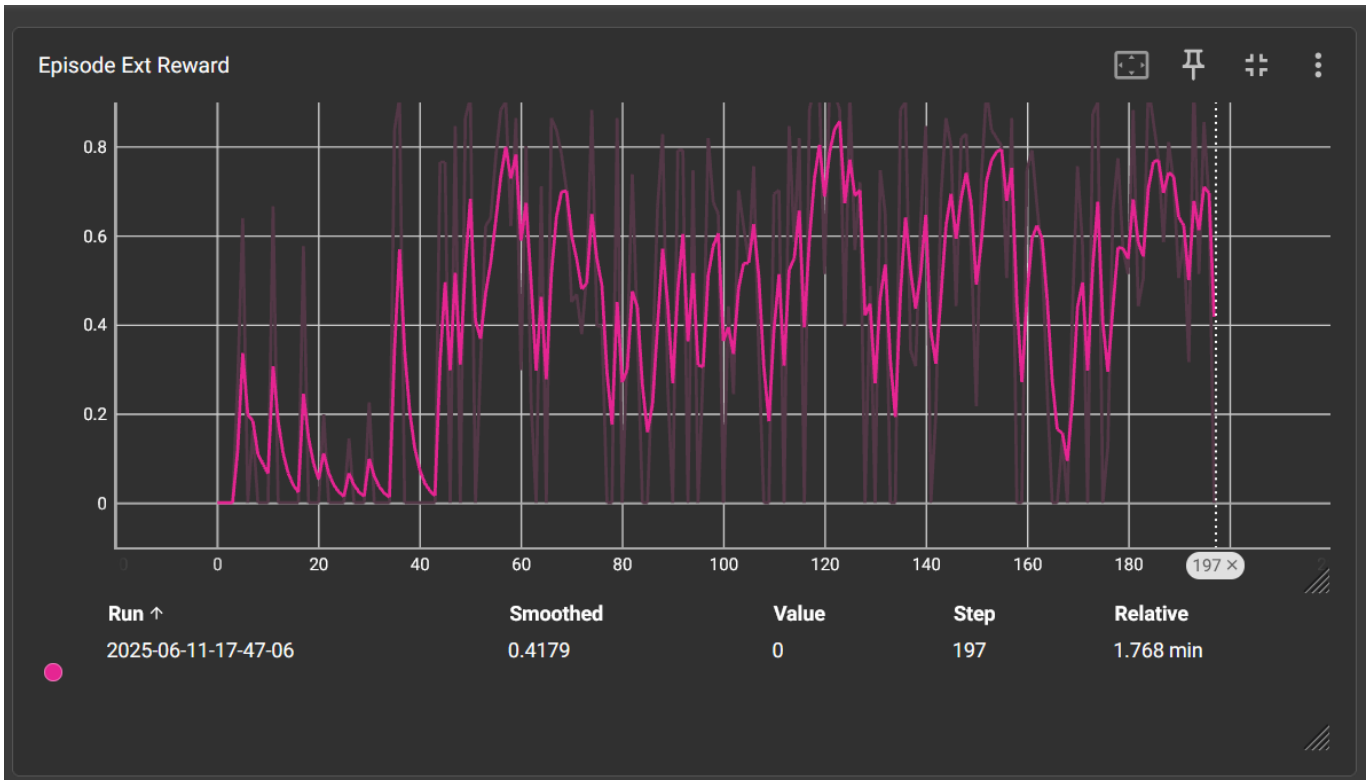

Figure 1: reward when `int adv coeff` = 1

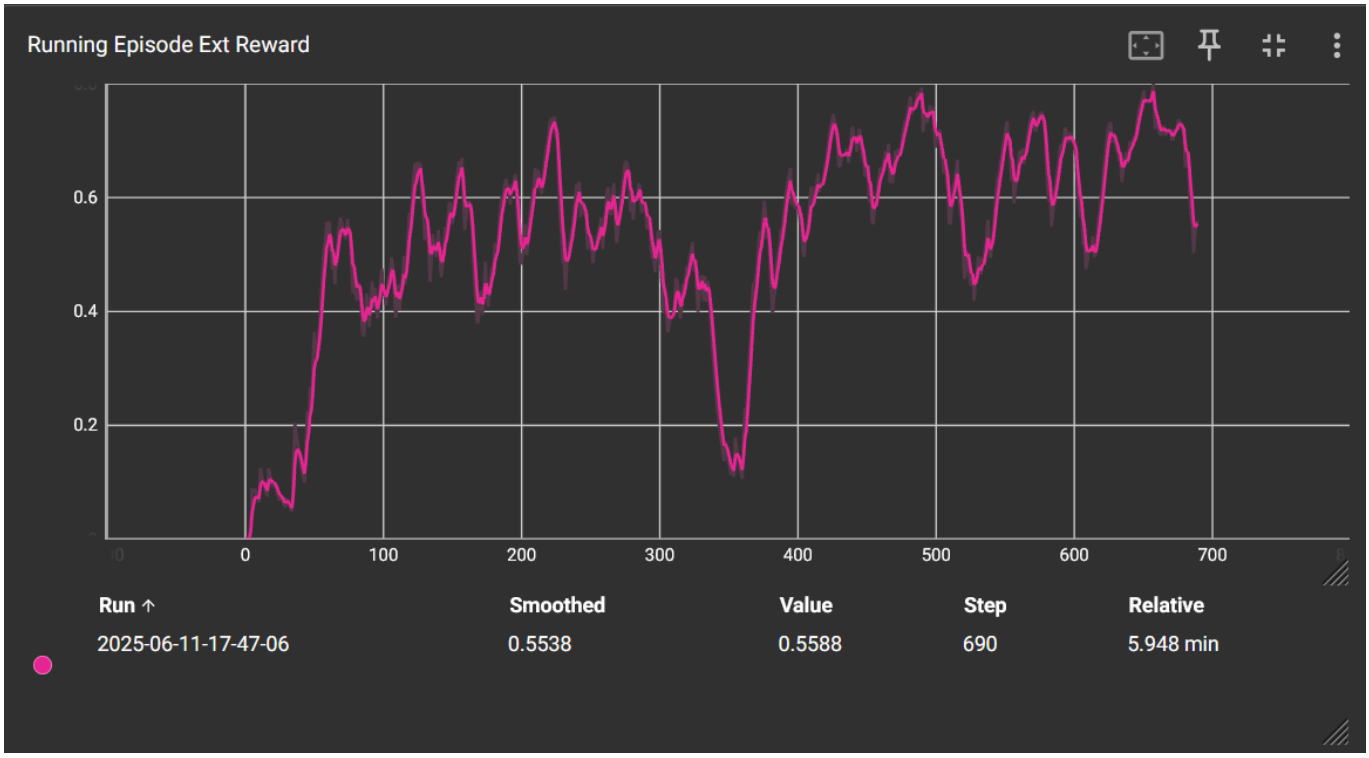

Figure 2