Task 2 Answers

Declaration: I used ChatGPT to enhance my writing and extend my answers.

Q1: The core idea behind Random Network Distillation (RND) is to encourage an agent to explore novel states by creating its own curiosity-driven reward signal. Imagine two neural networks: one is a fixed, randomly initialized "target" network that never gets trained, and the other is a "predictor" network that tries to guess the output of the target network for a given state. When the agent encounters a state it has seen many times before, the predictor network will become very good at mimicking the target network, resulting in a low prediction error. Conversely, when the agent visits a new, unfamiliar state, the predictor will struggle to make an accurate guess, leading to a high prediction error. This prediction error itself is then used as an intrinsic reward; by rewarding the agent for visiting states with high prediction error, RND effectively encourages it to seek out novelty and systematically explore unfamiliar parts of the environment, which is especially useful when external rewards are sparse or non-existent.

Q2: Using both intrinsic and extrinsic returns in the PPO loss function allows the agent to successfully balance two crucial goals: exploring its environment and accomplishing the actual task.

Extrinsic returns, which come from the environment's true rewards (like reaching a goal), guide the agent toward the ultimate objective. However, in environments where these rewards are rare, an agent relying only on them might never stumble upon a reward and would fail to learn anything useful. This is where intrinsic returns, generated by RND as a "curiosity" bonus for visiting novel states, become vital. They provide a constant stream of motivation for the agent to explore, preventing it from getting stuck and helping it discover the rare but essential extrinsic rewards more efficiently. Combining them creates a more robust agent that explores intelligently while still being focused on solving the task it is meant to perform

Q3: increasing the predictor_proportion makes the prediction problem for the Random Network Distillation (RND) module more difficult, as more features are used in the RND loss calculation.

This can have two opposing effects on learning:

- It might help learning by creating a more sensitive and longer-lasting novelty signal. A more complex task means the predictor network takes longer to accurately mimic the target network, potentially encouraging more thorough exploration before the intrinsic reward disappears.

- It might hurt learning if the prediction task becomes too difficult. This could lead to a noisy and unstable intrinsic reward signal that provides no useful gradient for exploration, effectively confusing the agent and destabilizing the overall training process.

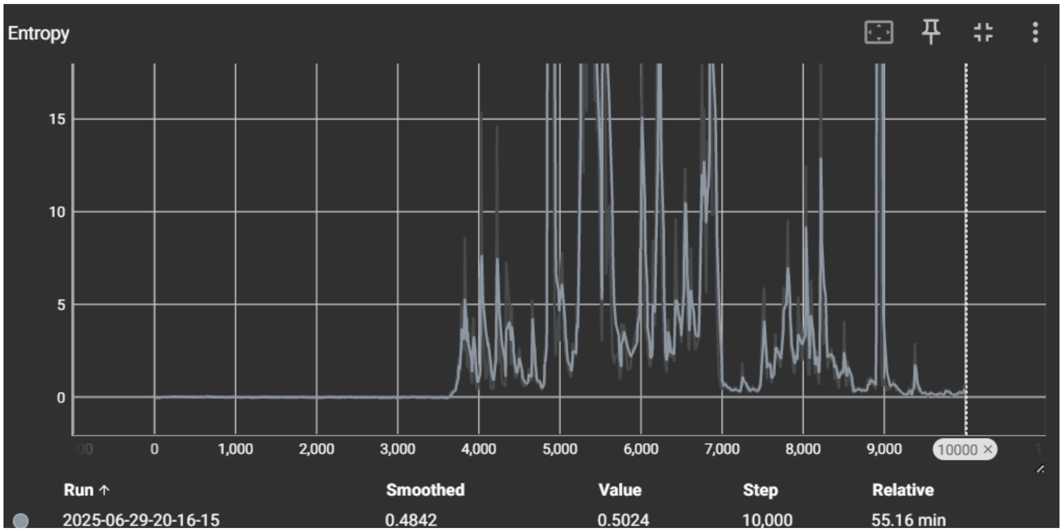
Therefore, the effect depends on finding the right balance; it is a hyperparameter that would need to be tuned for optimal performance.

Q4: setting the intrinsic advantage coefficient to 0 effectively removes the curiosity-driven exploration bonus from Random Network Distillation (RND).

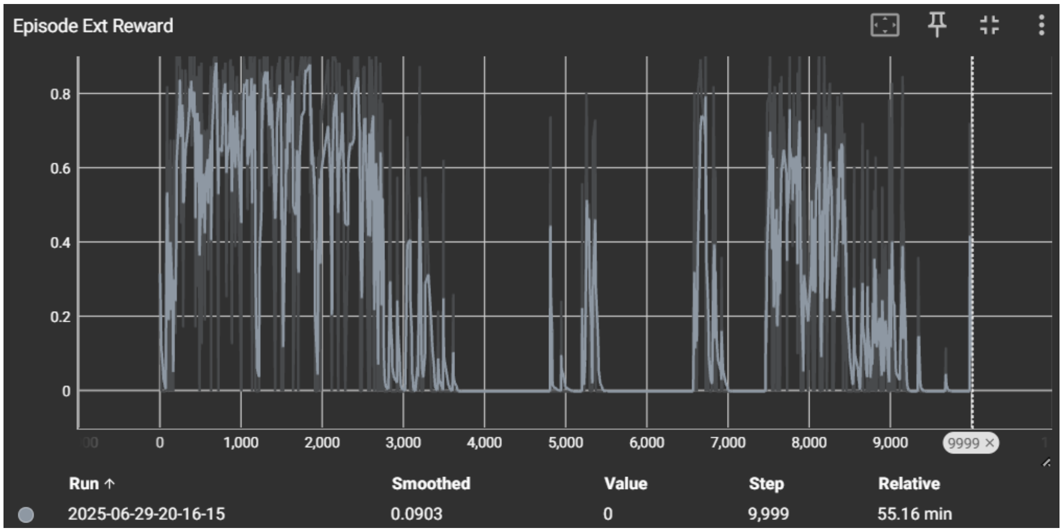
Without intrinsic motivation, the agent's behavior would change drastically. It would no longer be rewarded for exploring novel states and would rely solely on the environment's external (extrinsic) rewards. In a sparse-reward setting environment, the agent would likely exhibit poor exploration behavior. It would struggle to discover the goal, leading to a significant drop in performance and a failure to learn an effective policy, resulting in minimal to no extrinsic rewards being collected.
Here the environment is small, so this does not happen. In this problem intrinsic rewards decrease as we mentioned, and model explores less. But the performance doesn’t worsen drastically.

Q5: Initially, the intrinsic rewards remain low as the agent starts to explore and learn the most immediate states. The reward signal then increases as the agent's exploration becomes more effective, allowing it to discover a broader base of states. Finally, as these states become familiar and the predictor network learns to accurately anticipate the target network's output, the prediction errors decrease, causing the intrinsic rewards to decay. (Look at Int Value Loss tensorboard)

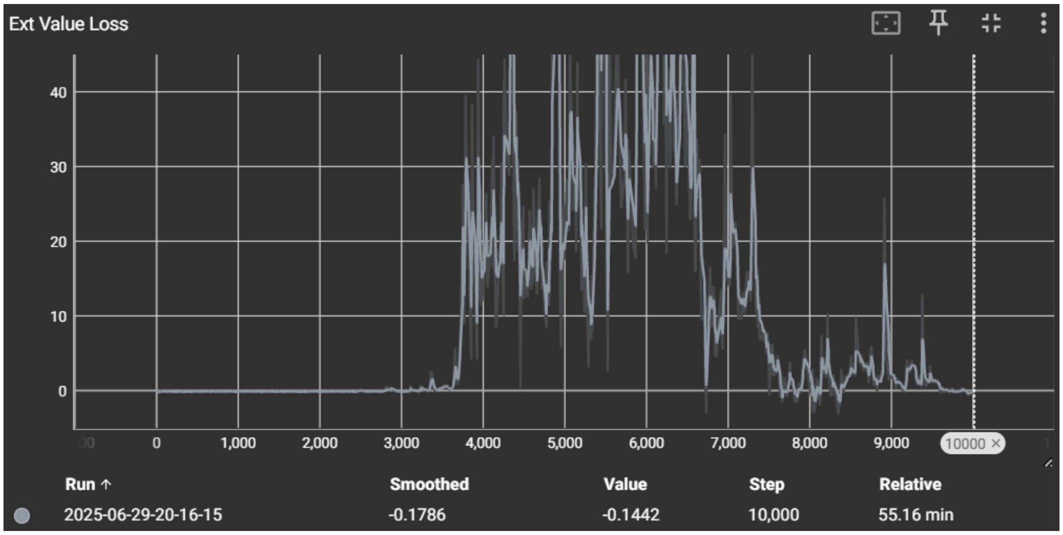
Tensorboards:



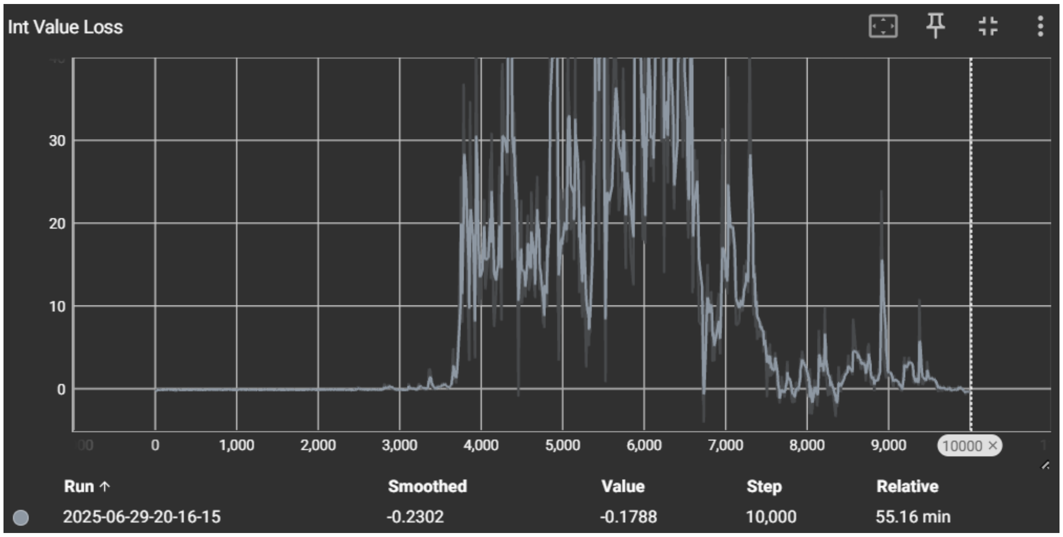
Run ↑	Smoothed	Value	Step	Relative
2025-06-29-20-16-15	0.4842	0.5024	10,000	55.16 min



Run ↑	Smoothed	Value	Step	Relative
2025-06-29-20-16-15	0.0903	0	9,999	55.16 min

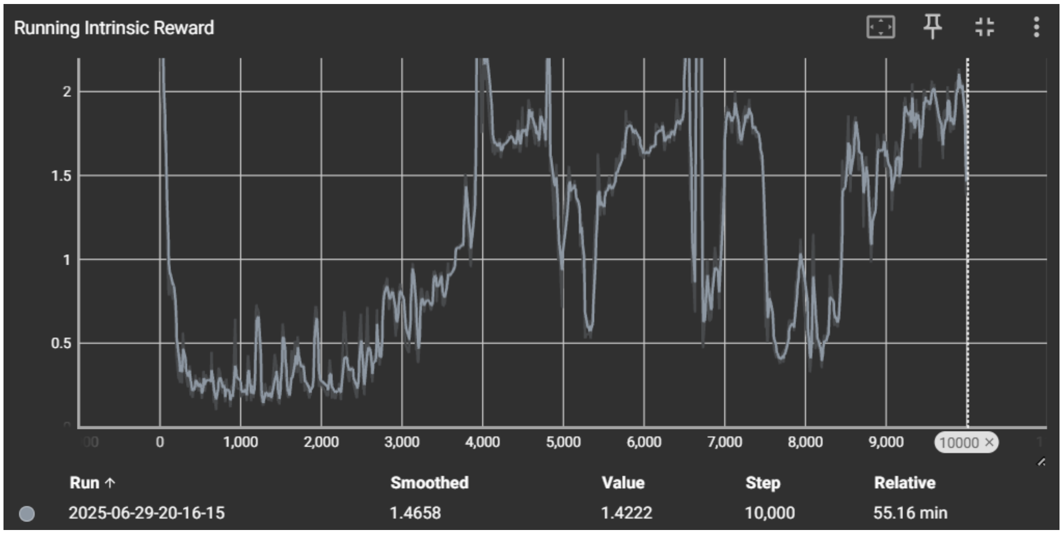


Run ↑	Smoothed	Value	Step	Relative
2025-06-29-20-16-15	-0.1786	-0.1442	10,000	55.16 min



Int Value Loss

	Run ↑	Smoothed	Value	Step	Relative
●	2025-06-29-20-16-15	-0.2302	-0.1788	10,000	55.16 min



Running Intrinsic Reward

	Run ↑	Smoothed	Value	Step	Relative
●	2025-06-29-20-16-15	1.4658	1.4222	10,000	55.16 min