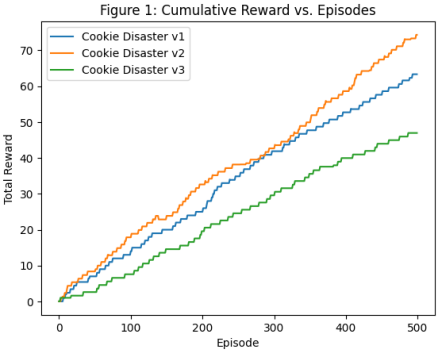
Part II - RL

In this report, we improve a patrolling agent to generalize across various environments. This task can be expressed as a Markov Decision Process (MDP) framework. The action space consists of accelerating either left or right, or not accelerating. Transition probabilities depend on the state and the robot's actions. The reward function provides incentives for collecting objects and penalties for crashing or failing. The discount factor which prioritizes short- or long-term rewards. And finally, the state space, which includes the robot's position and velocity, and the position and remaining time of falling objects, all of which are dynamic to the environment. Key challenges include the need for generalization across different environments, adapting to dynamic states, and handling partial observability with limited information. The agent must balance immediate rewards with long-term objectives, navigate a complex action space, and learn from sparse and delayed rewards. Addressing these challenges is essential for developing an adaptable agent.

To train an agent to maximize rewards across diverse environments, we used an actor critic implementation, which we decided to train on an abundance of generated environments. Our agent, the `CookieAgent`, was initialized with specific parameters, after with each new environment we updated environmental parameters that we needed to preprocess and normalize each state. This ensured that the agent would be able to navigate any environment regardless environment size, as the preprocessing transforms any state to be within [-1,1]. We decided to use environment sampling to train both on the provided environments from cookiedisaster as well as randomly creating environment configuration for the agent to train on. For each environment the agent runs 5 initializations before getting a new environment. This is so it has a chance to learn more from an environment, but hopefully changes early enough so it doesn't learn to overfit to this one environment. This is to enhance generalization, allowing the agent to train under various conditions.

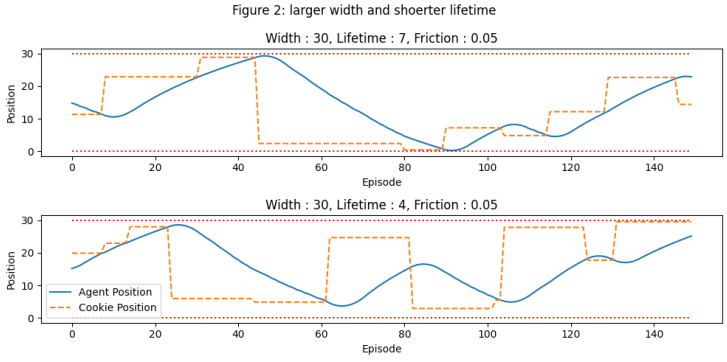
Our approach relies on the assumption that the environments share underlying structures the agent can exploit with only the config changing (width, lifetime, and friction function). Its effectiveness depends on having a diverse and representative sample of environments and a robust learned policy, emphasizing the importance of thorough testing across different scenarios. We used adaptive exploration strategies, like epsilon-greedy method, to balance exploration and exploitation, and introduced complexity gradually through varied environments. In addition, we would like to note that we have only trained our agent on environments of smaller sizes, and we are thus unsure of its effectiveness as the parameters grow larger. Our trained agent is also dependant on a new agent being initialized for each new environment it is tested on, as it uses the first call of the `select_action()`method to set the environmental parameters.

Figure 1: Cumulative Reward vs. Episodes



We analysed the results by looking at the agent's cumulative rewards across its run. Figure 1 shows our trained agent on the provided cookie environments, we can also see the agent always gathers positive rewards with little to no negative rewards. In general cookie disaster 2 does best. This is the environment with smallest width and lowest cookie lifetime, this does result in our agent colliding at times, either due to too high velocity or not hitting the walls, but as the cookies appear often and generally close by, it results in the best cumulative reward. Cookie environment 1 is a close second. While cookie 3, has both longer width and higher lifetime, the agent seems to slow down and prioritize not colliding at a high velocity instead of always going after highest reward, this in turn results in less collision, but also less cookies being gathered.

Figure 2: larger width and shoerter lifetime



In addition, we also tested it on some other configurations with higher widths and even lower lifetime and the agent does indeed select a more careful approach. This can be seen in Figure 2 with how the agent slows down before reaching the cookie. In addition, if the width is too large and lifetime too short it does accumulate some negative rewards, but still collects some a decent number of cookies.