Enzo Benoit-Jeannin

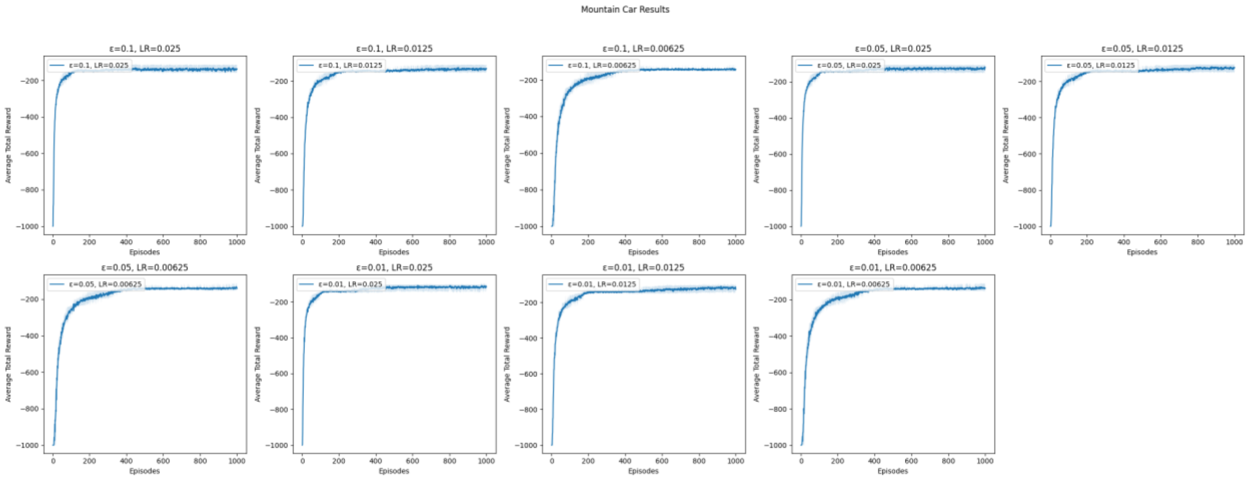
Sasha Denouvilliez-Pech

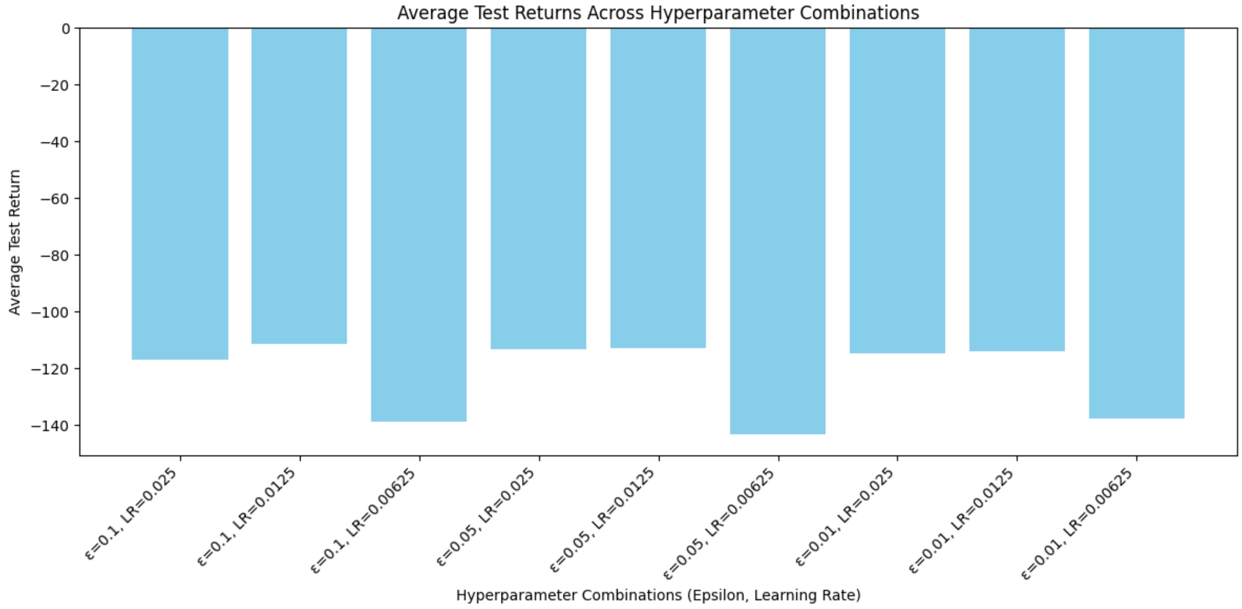
Assignment 3-Report

1: Value-based methods with linear function approximation

Mountain Car experiments

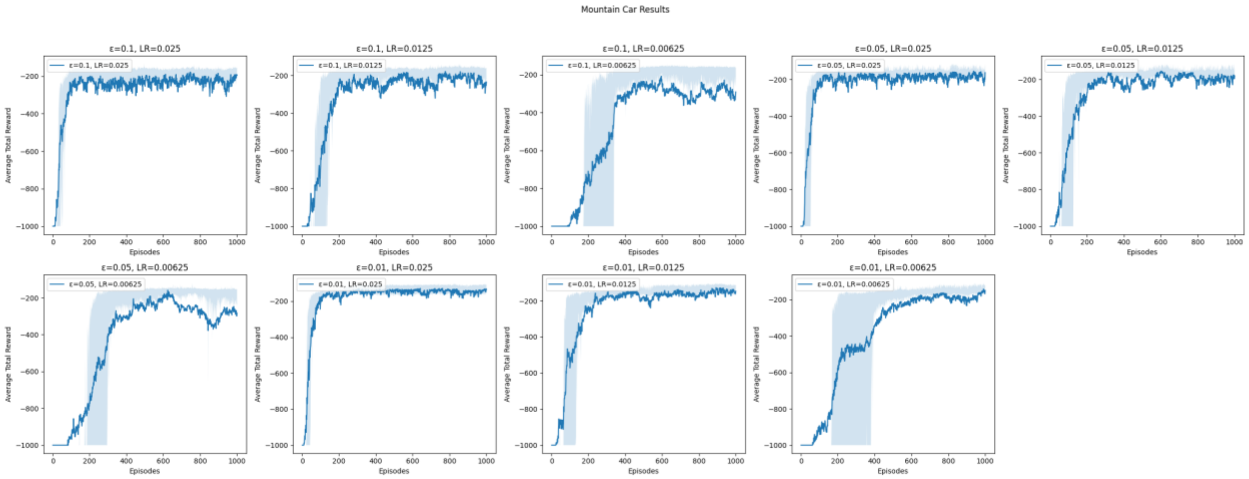
Q-Learning

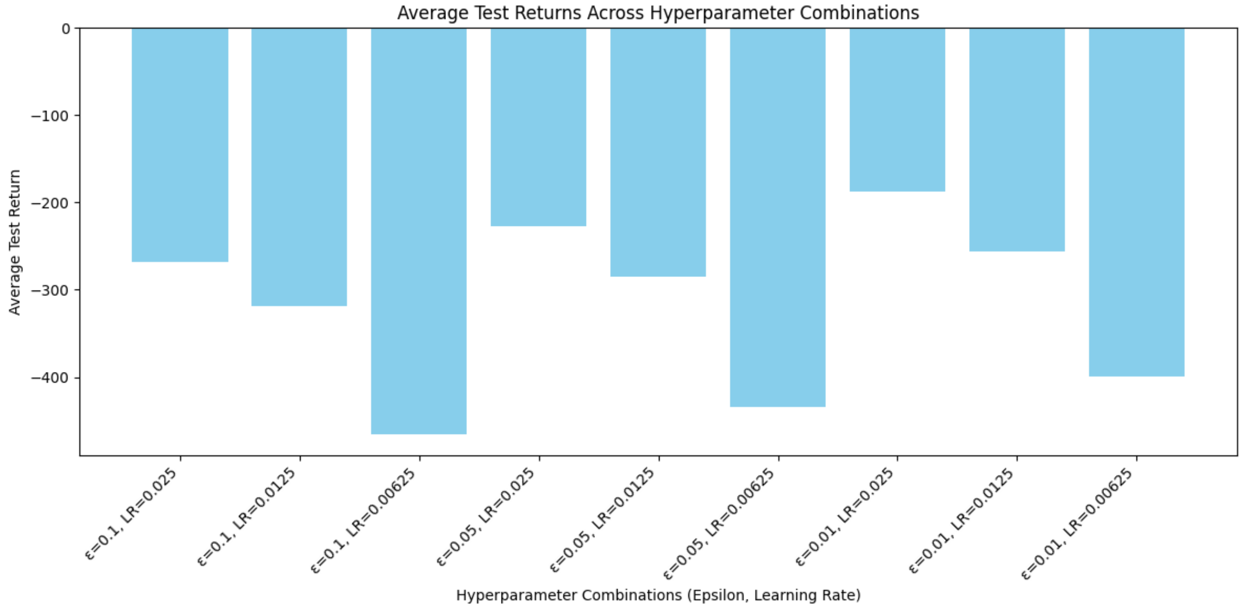




The Q-Learning algorithm seems to learn fairly well on the Mountain Car environment as can be shown by the training and test results. The maximum number of steps was set to 1000 to increase the amount of training per episode. The different hyperparameter combinations perform uniformly across both training and testing. It is worth noting slightly bigger learning rates, 0.025 and 0.0125, both lead to better testing rewards than using a learning rate of 0.00625.

Expected SARSA

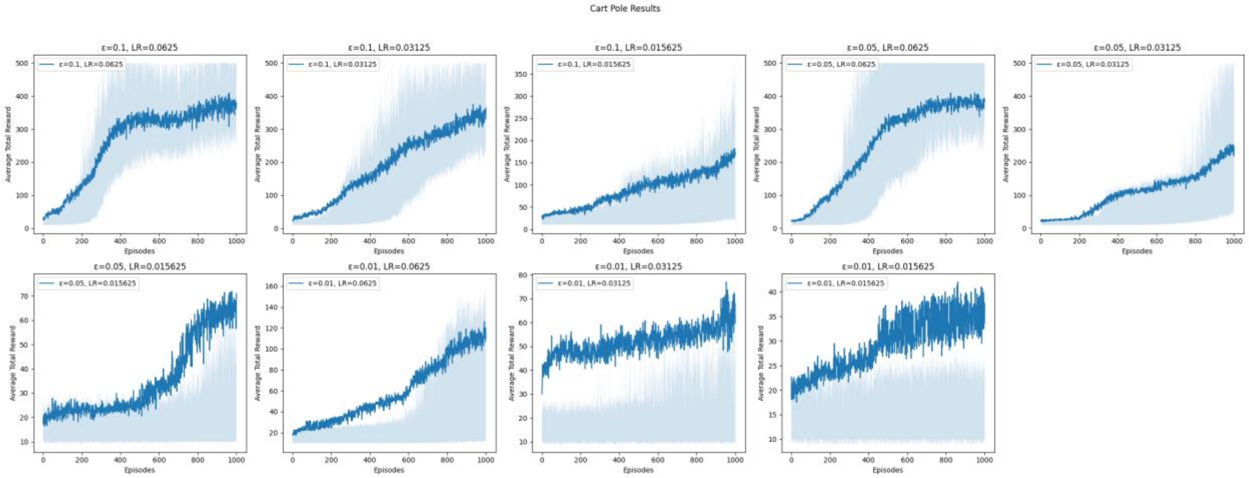


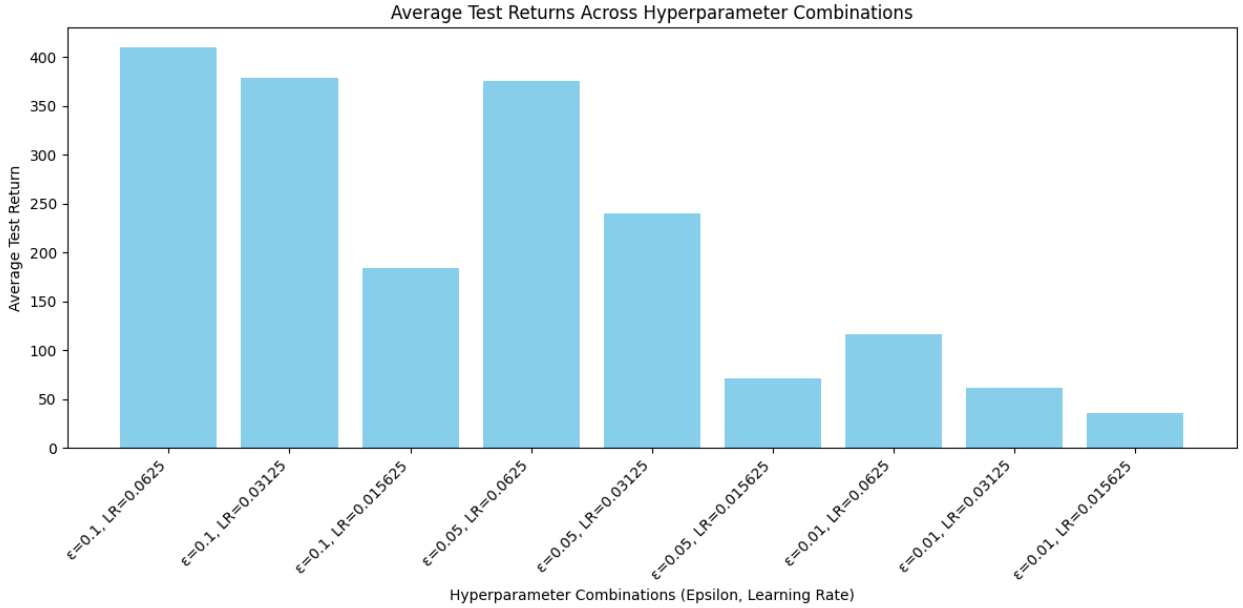


The Expected Sarsa algorithm learns well on the mountain car environment. The training results are much more volatile than for Q-Learning and also more conservative. It is also worth noting that learning is not as quick and smooth as it was for Q-learning. This is highlighted by the usual "jump" we can observe on the average total training reward suddenly increasing within the first 300 episodes. These jumps seem to be less consistent in terms of episode at which they occur with the 0.00625 learning rate as the interquantile range is wider indicating more variance in performance from one trial to the other. Test results are also much lower compared to Q-Learning. Interestingly, bigger learning rate and smaller epsilon perform better overall. As Q-learning tends to prioritize exploration more aggressively compared to Expected Sarsa, it could explain the difference in performance. However, Expected Sarsa seemed to not benefit from more exploration during training, i.e. higher epsilon.

Cart Pole Experiments

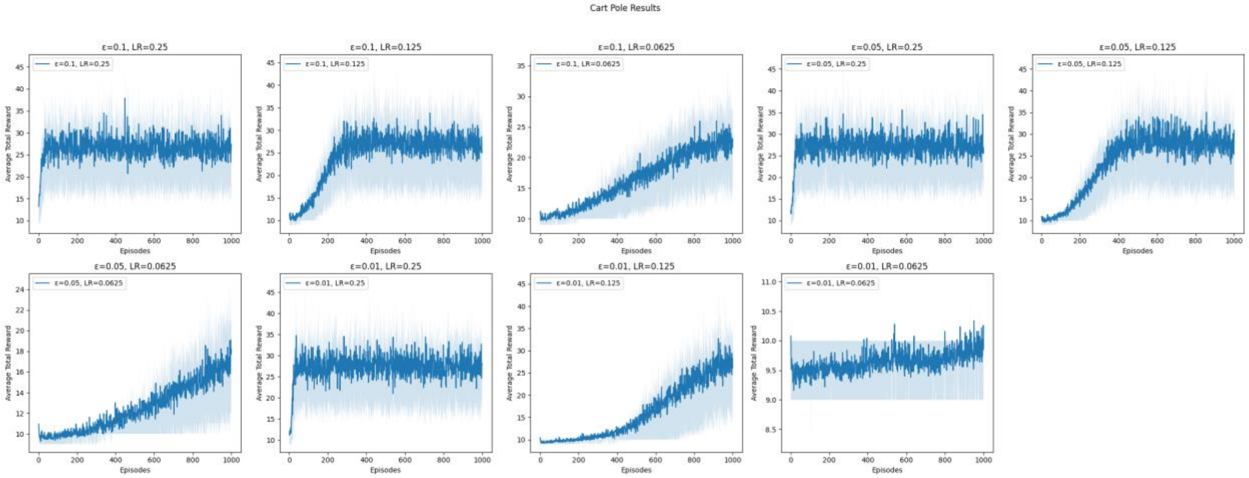
Q-Learning

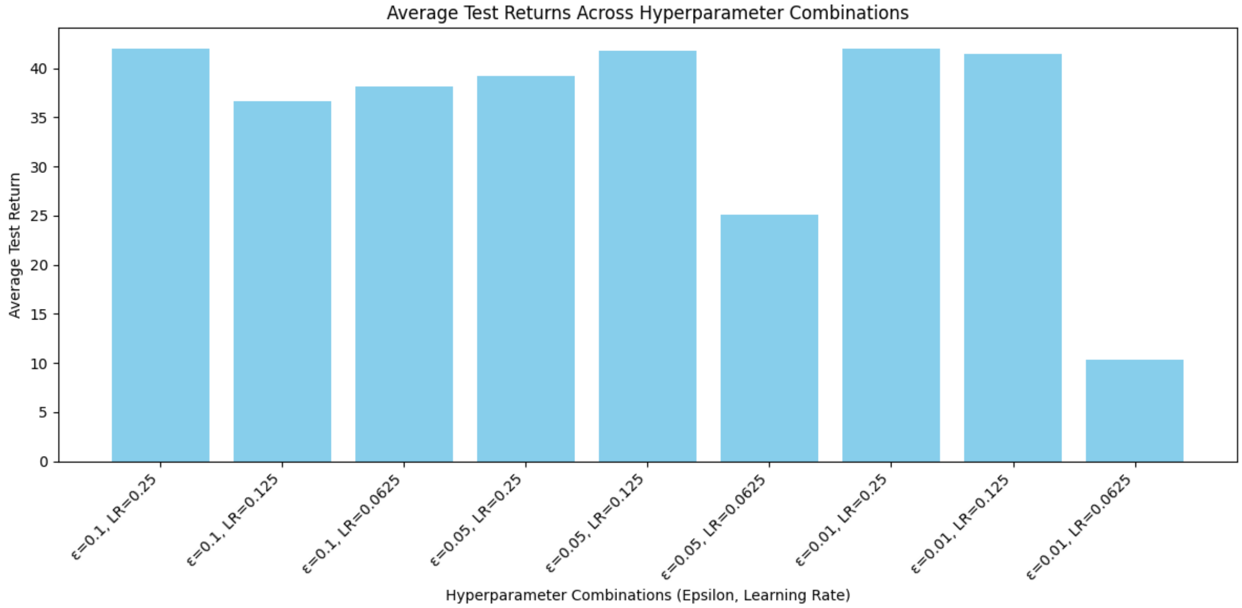




Again, the Q-Learning algorithm learns very well on the cart pole environment as can be shown by the training and test results. However, the results are much more volatile, maybe due to the more random nature of the problem or higher state dimension. On the training side, the interquantile range is wider and even out of the mean across all episodes. Additionally, the learning curves vary a lot with hyperparameter combination. The testing results suggest that more exploration, higher epsilon, leads to better exploitation and thus better average test returns. Moreover, slightly bigger learning rates, 0.025 and 0.0125, lead to better testing reward as well.

Expected Sarsa





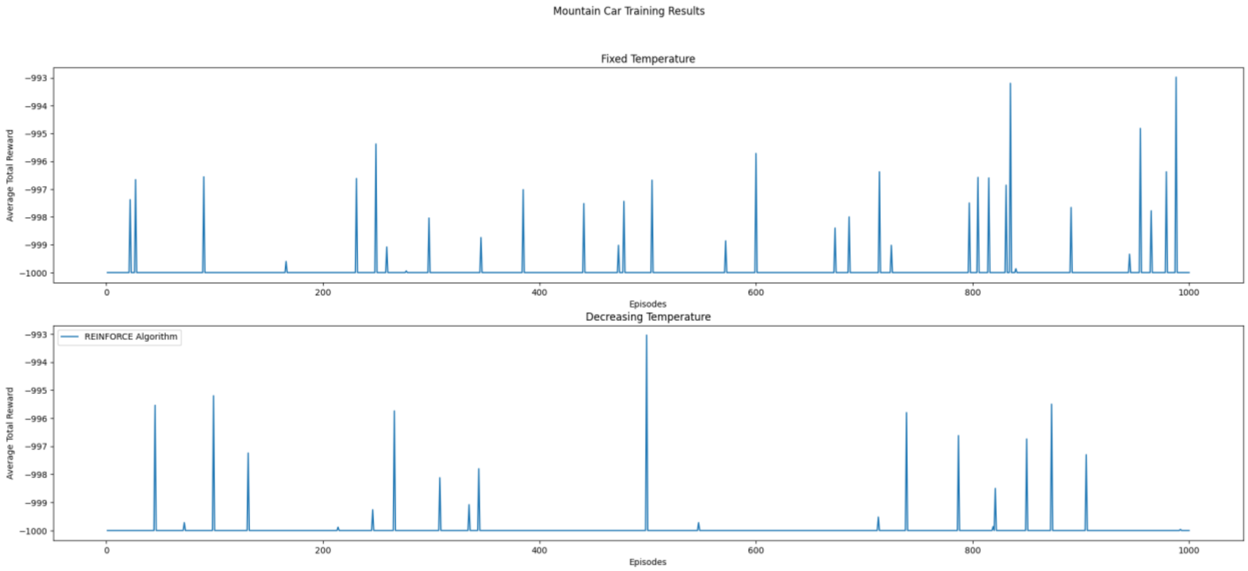
The Expected Sarsa algorithm learns poorly on the cart pole environment, up to 10 times worse than the Q-learning algorithm. For some hyperparameter combination, the algorithm gets stuck in a local optima and stops learning. Similarly to the mountain car environment, training performance is volatile as shown by the wide interquantile ranges. It is worth noting that epsilon = 0.01 and LR = 0.0625 does not learn at all. Test results show that bigger learning rates tend to perform a little bit better as mentioned above. Again, the lack of exploration from Expected Sarsa compared to Q-learning could explain why the algorithm does not converge properly. Also, considering the bigger state space of the cart pole environment, this behavior might be more penalized.

2: Policy Gradient Theorem

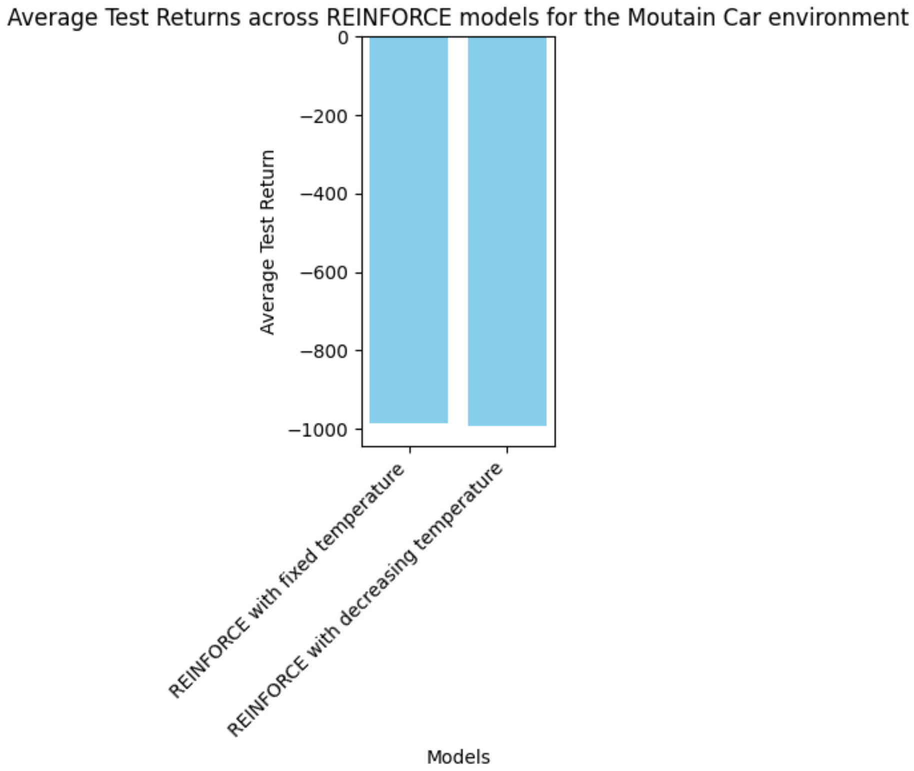
3: Policy-based methods with linear function approximation

Mountain Car experiments

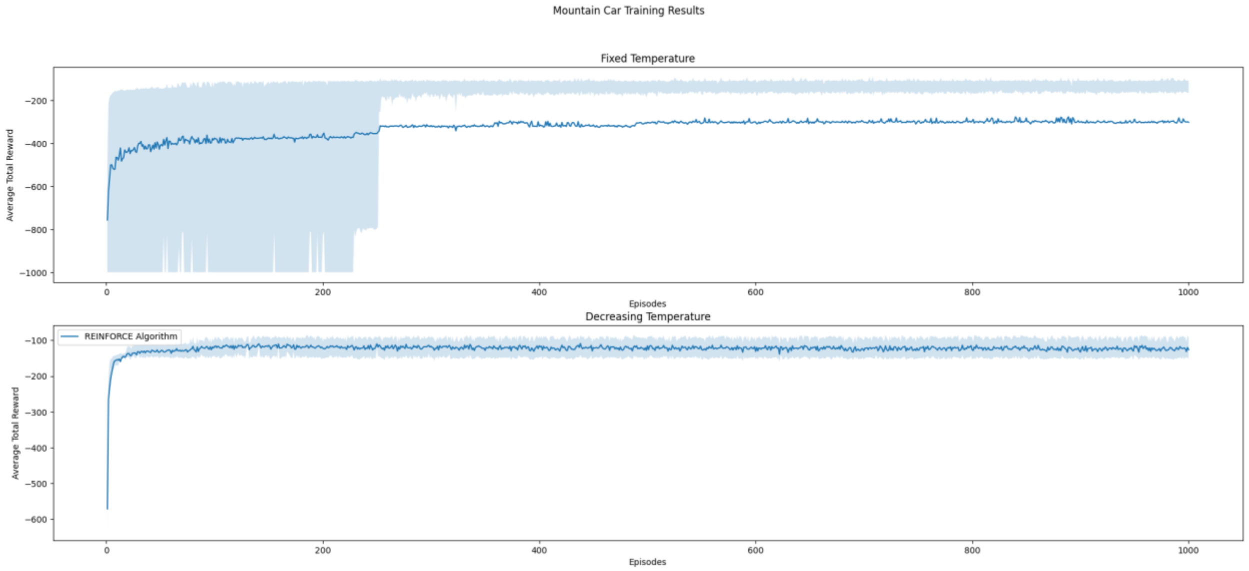
REINFORCE



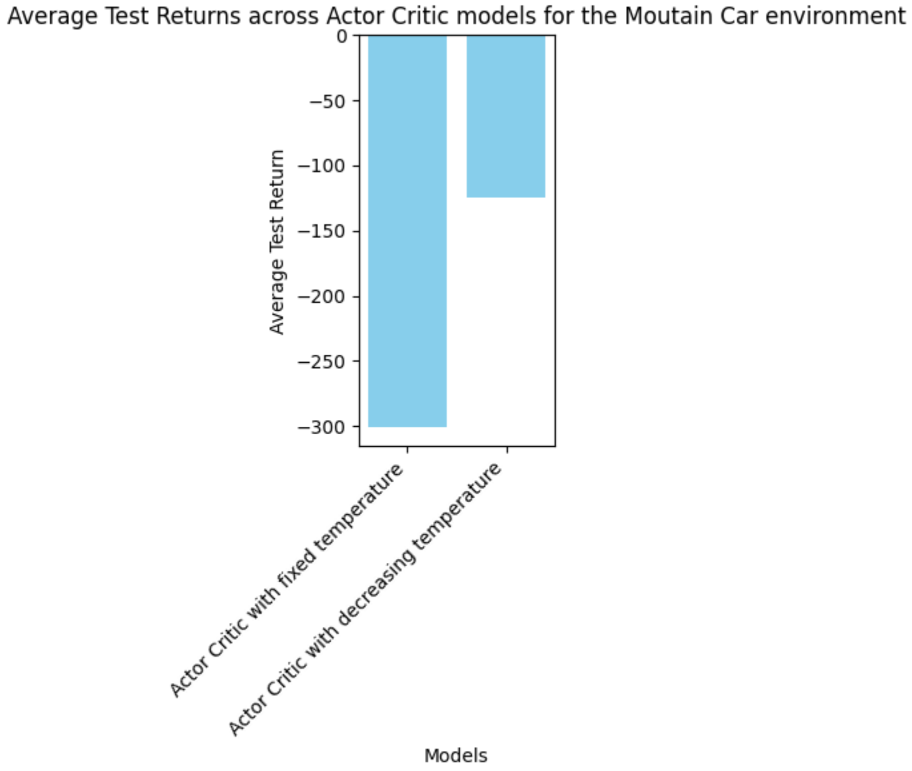
In the Mountain Car v0 environment, the REINFORCE algorithm's performance was notably poor with both fixed and decreasing temperatures. Despite some occasional spikes in reward, the overall trend was a flat line at the lowest possible average total reward of -1000. This indicates that the agent consistently failed to reach the goal within the allotted 1000 steps, receiving a -1 penalty for each step without success.



A2C



Analyzing this resulting graph, the Actor-Critic algorithm's performance in the Mountain Car environment varies with the choice of the temperature parameter. With a fixed temperature of 0.1, the algorithm demonstrates different learning times within the initial 210 episodes over all 50 trials, as indicated by the interquantile ranges which suggest high inconsistencies in the rewards obtained during training at the beginning. In contrast, the approach of decreasing the temperature throughout the episodes, we notice that the algotithm learns in a more stable way across all trials (so the interquantile range is smaller) and the rewards are higher. This suggests that the decreasing temperature approach is more effective as it leads to more consistent and higher rewards.



Results Analysis

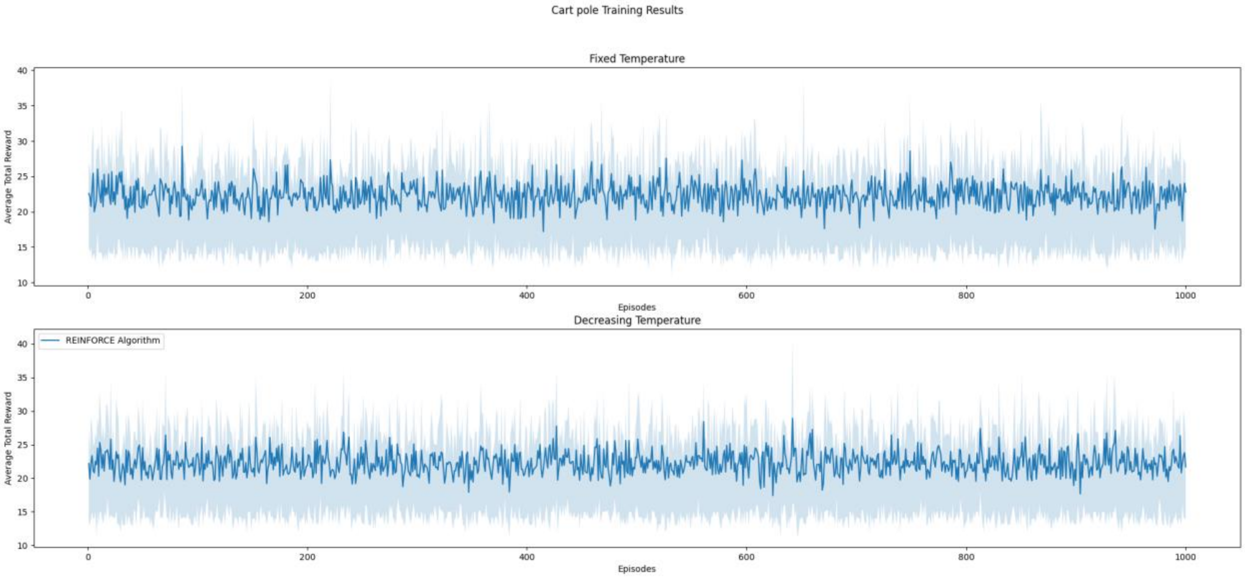
First we notice that the REINFORCE algorithm generally performed very poorly on the Mountain Car v0 environment, both when fixing the temperature or decreasing it. Indeed, although we can notice some spikes in the average reward during training, both REINFORCE algorithms always get an average total reward of -1000, which is the worst possible score, as the environment provides a -1 reward for every time step the agent did not reach the flag (in this case the maximum number of time steps was set to 1000). Moreover, on testing results both REINFORCE algorithm. On the other side, the Actor Critic algorithm performed significantly better than the REINFORCE algorithm, both with fixing the temperature and decreasing it. We note that decreasing the temperature with the number of time steps drastically improved the performance of the algorithm compared to just fixing it. Indeed, with a fixed temperature, the average test return is around -300 while decreasing the temperature increases that average test reward to around -120.

The Actor-Critic method outperforms REINFORCE in the Mountain Car environment due to its greater sample efficiency. This efficiency stems from the algorithm's ability to update both its policy (the actor) and value function estimates (the critic) incrementally with each step of an episode, leveraging bootstrapped estimates for more frequent and relevant adjustments. On the other hand, REINFORCE updates the policy only at the episode's end, which can hinder learning in environments characterized by longer episodes.

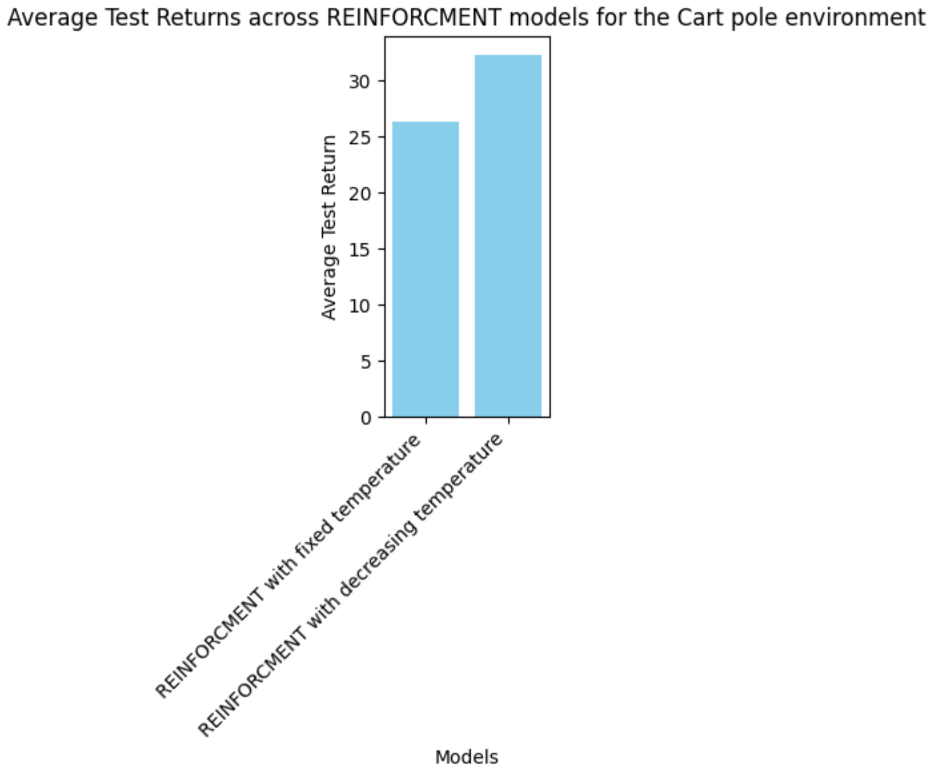
Moreover, REINFORCE relies on the cumulative return for policy updates which can introduce high variance, as the total rewards from an episode can fluctuate greatly—especially in a scenario like Mountain Car, where successfully reaching the flag vastly impacts the return. Actor-Critic, on the other hand, benefits from lower variance in updates. It achieves this by bootstrapping from the critic's value estimates, thus providing a more consistent and reliable signal for learning. This fundamental difference in approach enables Actor-Critic to learn more stable and effective policies faster than REINFORCE.

Cart-Pole experiments

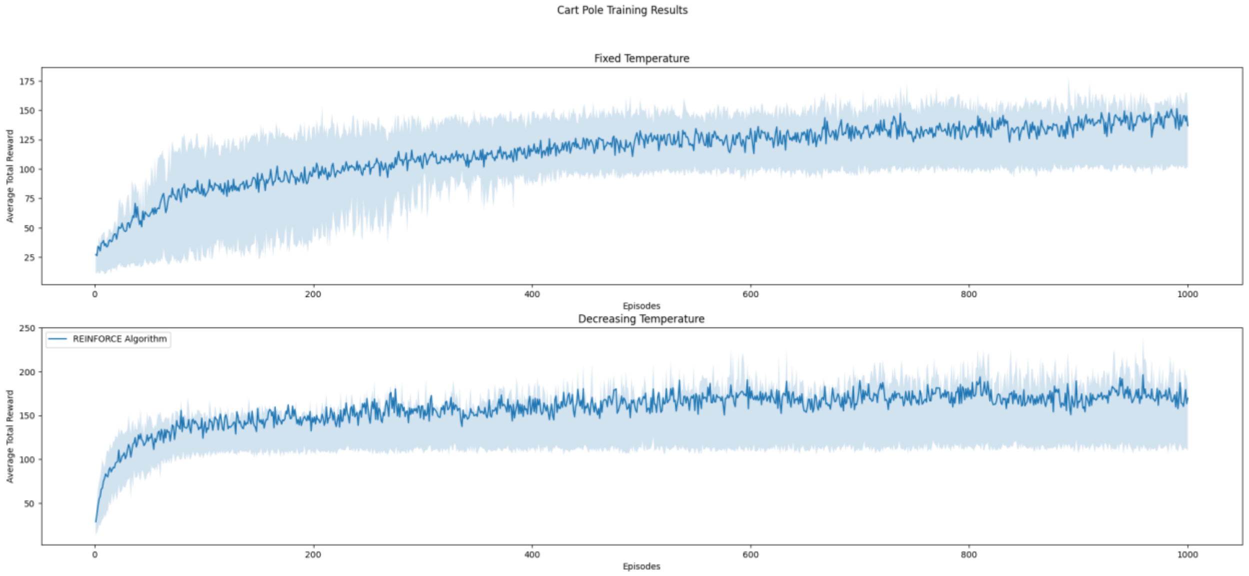
REINFORCE



In the CartPole v1 environment, the REINFORCE algorithm seems to exhibit high instability in training rewards, as shown by the variance seen in the above graph. The average reward rarely goes above 35, which implies that the model struggles to learn the task effectively. In the CartPole scenario, the agent receives a reward for each timestep the pole remains upright on the cart. Thus, the total reward corresponds to the number of timesteps before the pole falls. The limited reward ceiling indicates that the algorithm fails to consistently find or maintain a strategy for balancing the pole for an extended period, terminating episodes prematurely when the pole deviates significantly from the vertical.



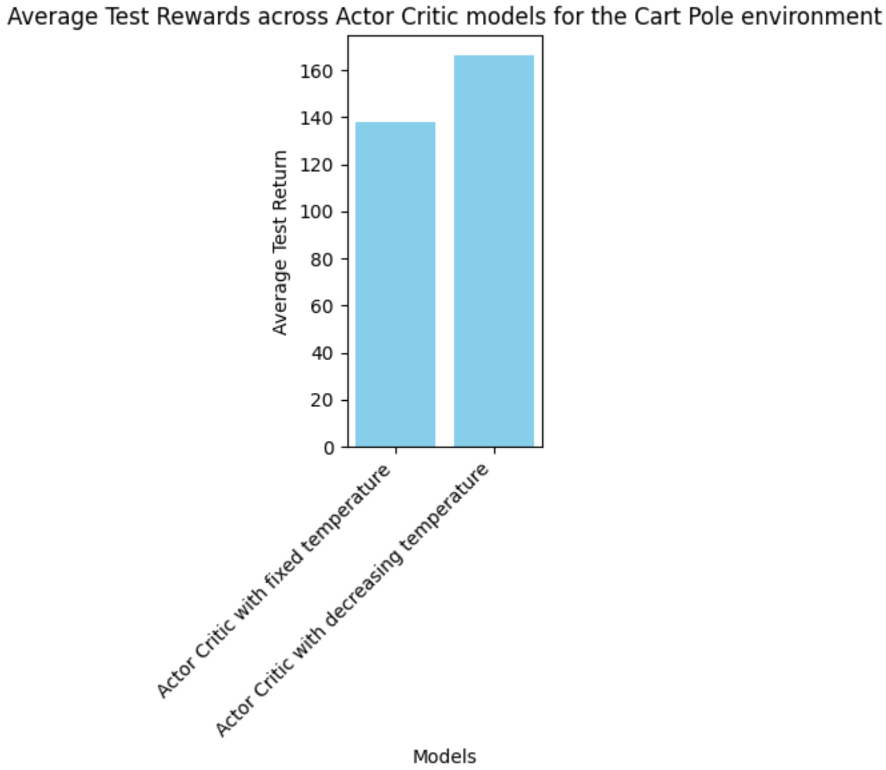
A2C



From these results, we observe an upward trend in the average training reward over episodes for both fixed and decreasing temperature settings. This gradual increase indicates the algorithm's learning progression with noticeable improvements in balancing the pole on the cart unlike the REINFORCE algorithm.

In the case of a fixed temperature, the average reward displays more volatility, as seen in the broader interquantile range. Conversely, when employing a decreasing temperature, the algorithm exhibits less volatility, indicated by a narrower interquantile range. This reduction in variability suggests a more stable learning process. As the temperature lowers, the policy shifts focus from exploration towards exploitation, consistently refining the strategy to keep the pole balanced.

Comparing the two temperature strategies, the decreasing temperature approach clearly leads to higher rewards. Notably, we can notice that using a fixed temperate shows a constant and slow increase in average total reward where decreasing the temperature shows a more rapid increase in average total reward which then plateaus around the 600th episode. This suggests that the decreasing temperature approach is more effective as it leads to more consistent and higher rewards. This conclusion is smoewhat similar to the MoutainCar environment one.



Results Analysis

In the CartPole v1 environment, we see a drastic difference in performance between the REINFORCE algorithm and the Actor-Critic method. REINFORCE struggles with unstable training rewards and doesn't manage to push the average training reward much past 35, indicating difficulty in learning to hold the pole on the cart. On the other hand, Actor-Critic shows improvement over time, with both fixed and decreasing temperatures leading to better results. The method is more stable, particularly with a decreasing temperature resulting in higher and more consistent rewards.

In both types of algorithm, we also note that the average testing rewards were always higher when using the decreasing temperature method over the fixed one. Indeed, in the REINFORCEMENT algorithm the test rewards reached 35 using the decreasing temperature method against 26 with a fixed temperature. Actor Critic also performed much better using the decreasing temperature method yielding an average test return of around 162 against 139 for a fixed temperature.

REINFORCE didn't perform well in the CartPole v1 environment, and its difficulties can be linked to the same issues that impacted its performance in Mountain Car. Each time the pole falls, the episode ends, and REINFORCE is expected to update its policy based on the total cumulative return of the episode. However, this approach introduces a high variance in the policy updates because the rewards can be highly inconsistent from one episode to the next—especially in CartPole, where each episode's length (and thus total reward) varies significantly depending on how long the pole is kept upright. This variance can make it challenging for REINFORCE to learn a stable and effective strategy. The algorithm essentially gets a mixed signal about which actions are truly beneficial for keeping the pole balanced. In contrast, Actor-Critic methods provide more regular and informative updates. They use the critic's value estimates to provide a steadier learning signal—through bootstrapping—and adjust the policy at every step rather than waiting until the end of the episode. This allows for a more nuanced adjustment to the policy, leading to faster and more stable learning.