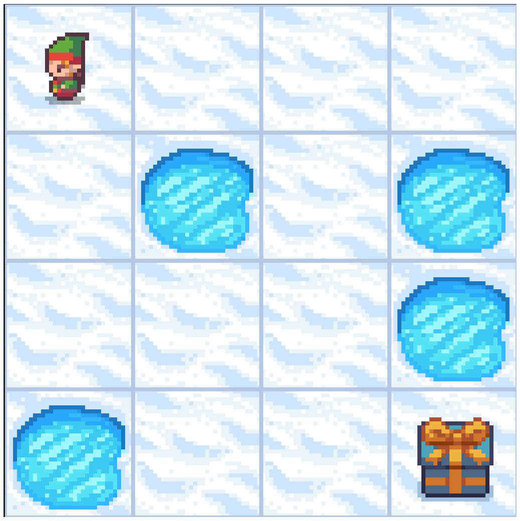
Reinforcement Learning Task
Giorgos Sapountzakis

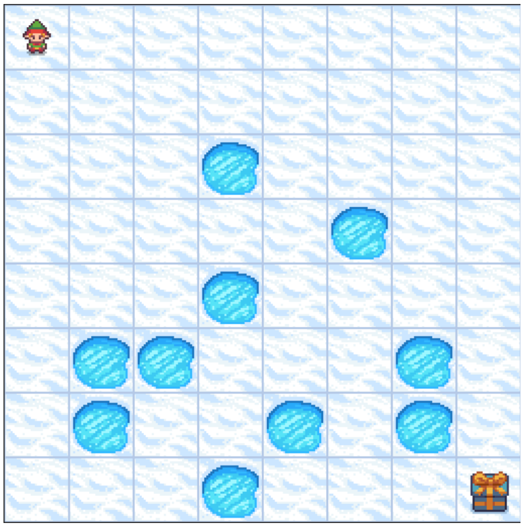
Overview:

Train an RL agent to navigate through a slippery frozen lake in order to reach the goal safely. The agent must learn to avoid holes and reach the goal using the least number of steps possible.

Implementation:

Following the instructions I used OpenAIGym (gymnasium) library for the enviroment setup and the stable baselines3 library for the RL algorithms used to train the agent and the final evaluation. The following two images showcase the enviroment that I used with the two different map sizes:





Procedure:

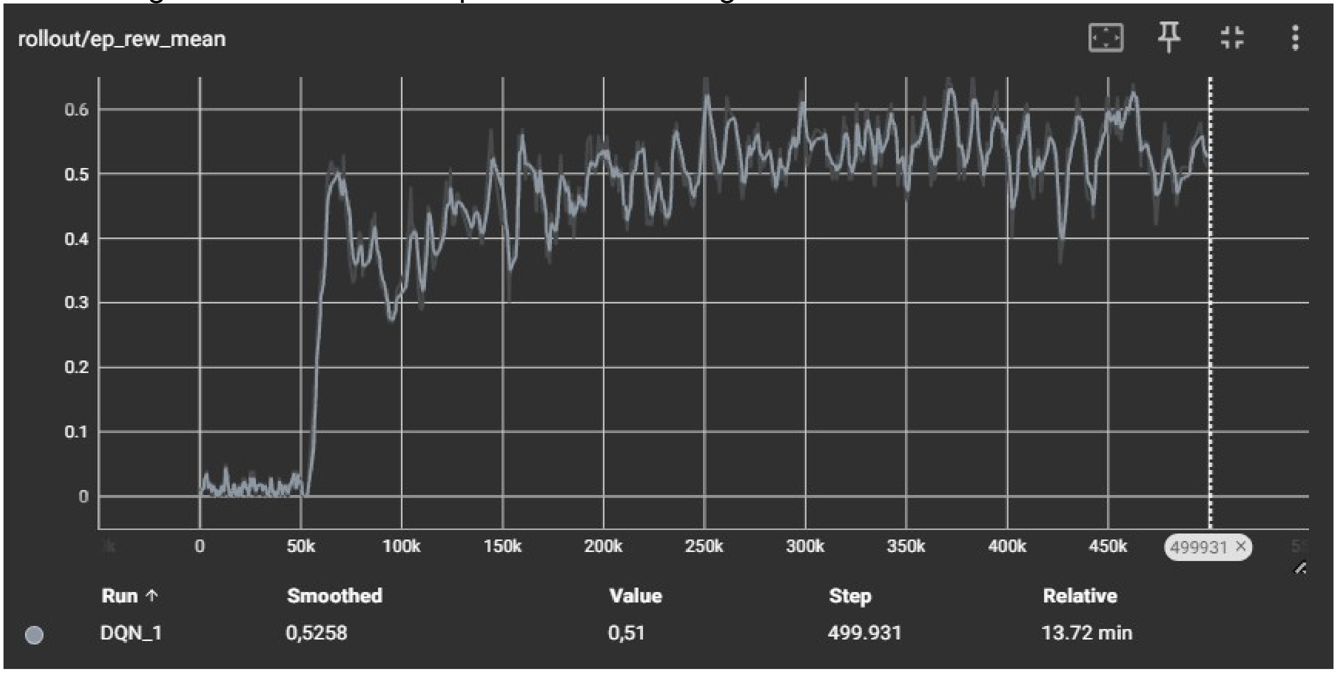
- Enviroment setup with map size = 4x4
- Agent initialization and training with DQN algorithm.
- Training curves display, training evaluation based on mean episode reward and standard deviation.
- Agent initialization and training with PPO algorithm.
- Training curves display, training evaluation based on mean episode reward and standard deviation.
- Same procedure for map size= 8x8
- Results comparison between different algorithms and map sizes.

Results presentation:

Map size= 4x4:

- Action space= Discrete(4)
  Agent can move only one block in the directions north ,south ,west ,east from his position.
- Observation space= Discrete(16)
  Whole enviroment space, consisting of 16 blocks.
- Reward range (0,1)
  An action, depending on the outcome rewards the agent in the range 0,1

For training with 500000 timesteps with the DQN agent:

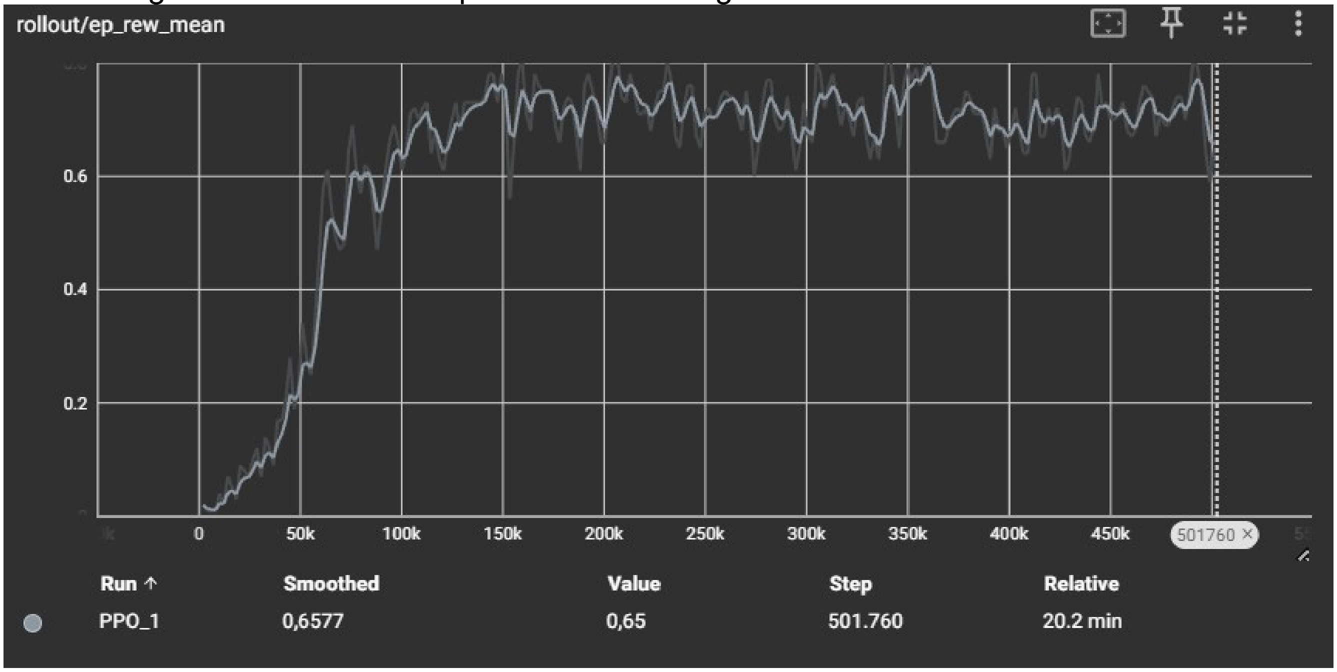


Run ↑	Smoothed	Value	Step	Relative
● DQN_1	0,5258	0,51	499.931	13.72 min

The evalutaion for the DQN agent:

- Mean episode reward: 0.726
- Standard deviation reward: 0.44600896851969246

For training with 500000 timesteps with the PPO agent:



Run ↑	Smoothed	Value	Step	Relative
● PPO_1	0,6577	0,65	501.760	20.2 min

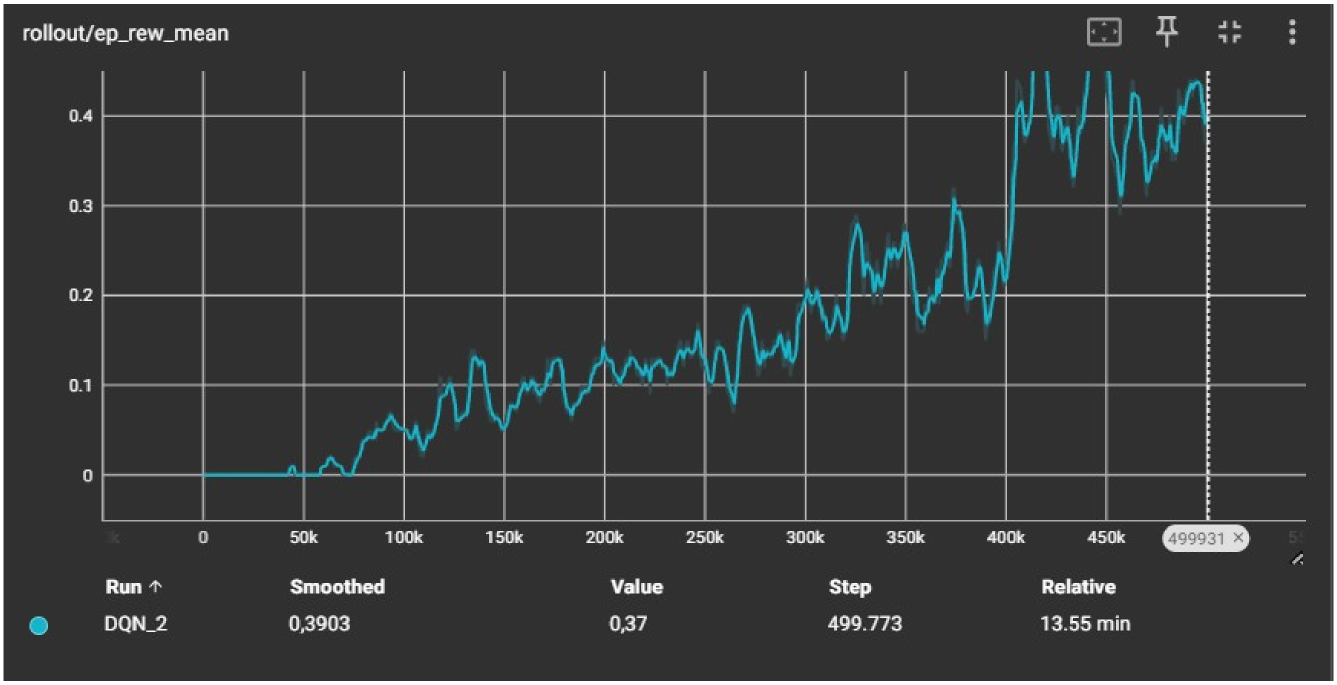
The evalutaion for the PPO agent:

- Mean episode reward: 0.739
- Standard deviation reward: 0.43917991757365227

Map size= 8x8:

- Action space= Discrete(4)
  Like before, the agent can move one block in the front, backwards and sideways, independently of the map size.
- Observation space= Discrete(16)
  Now we have a bigger space, consisting of 64 blocks.
- Reward range (0,1)
  Same as before, independent of the map size.

For training with 500000 timesteps with the DQN agent:

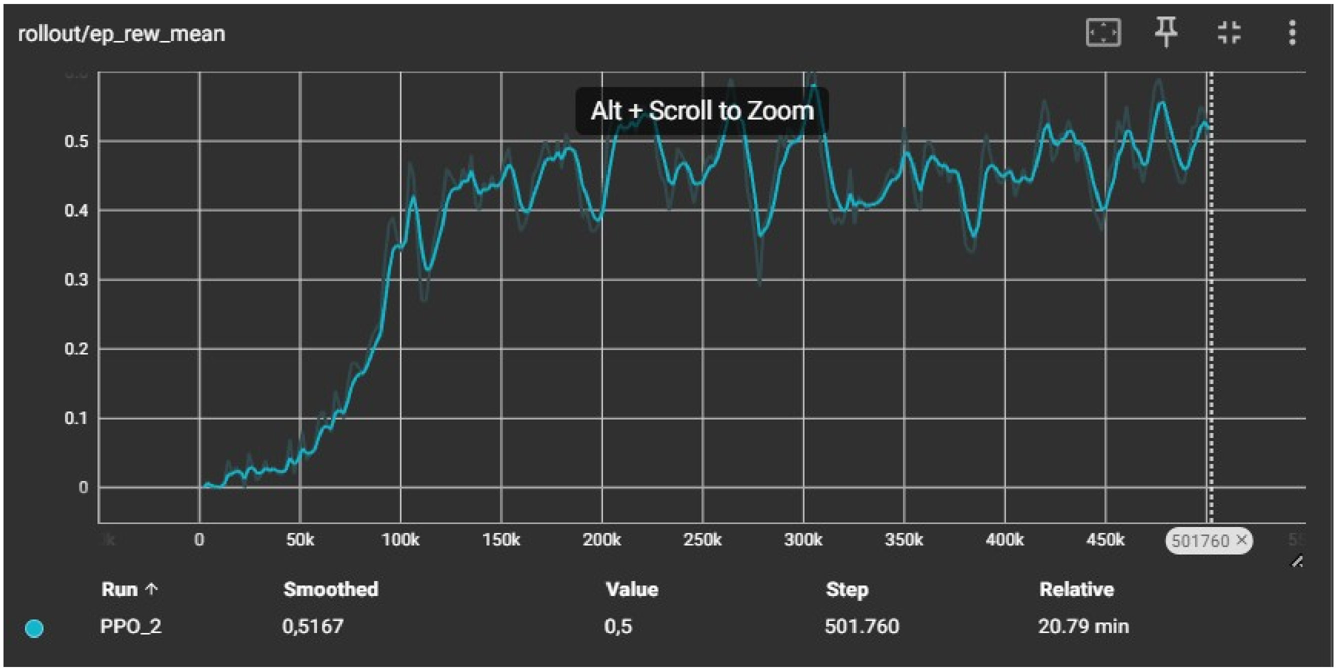


	Run ↑	Smoothed	Value	Step	Relative
●	DQN_2	0,3903	0,37	499.773	13.55 min

The evalutaion for the DQN agent:

- Mean episode reward: 0.537
- Standard deviation reward: 0.49862912068991716

For training with 500000 timesteps with the PPO agent:



	Run ↑	Smoothed	Value	Step	Relative
●	PPO_2	0,5167	0,5	501.760	20.79 min

The evalutaion for the PPO agent:

- Mean episode reward: 0.555
- Standard deviation reward: 0.4969657935914704

Observations:

There are a few observations that we can make after the end of our experiments. Both algorithms perform well for our problem and achieve very similar results, although there is a noticable difference in the training time as the PPO’s time was almost double than DQN’s. Also another observation we must make is the drop in the mean episode reward compared to the map size, as we see that when the enviroment got bigger the value dropped considerably. More plots and metrics can be found in the .ipynb files attached.

Conclusion:

The assigment was a very interesting problem and a nice introduction to the RL world and its usefull libraries like gymnasium and stable-baselines3. The experiments performed allowed me to use those libraries in practise and see how an agent works in a RL enviroment. It was a great experience overall.