FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

# CUSTOMIZING OPENAI GYM ENVIRONMENTS AND IMPLEMENTING REINFORCEMENT LEARNING AGENTS WITH STABLE BASELINES

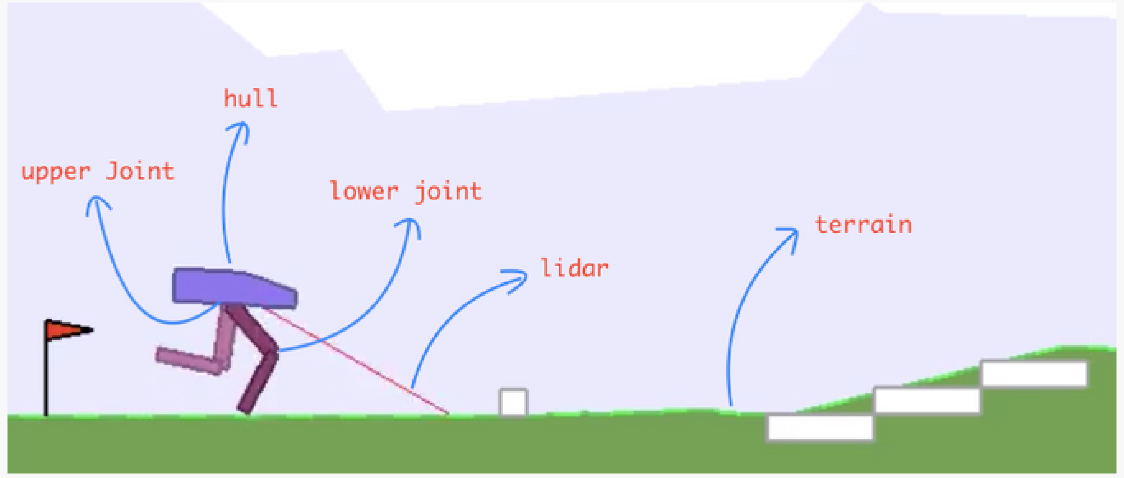
Introduction to Intelligent and Autonomous Systems

**DEVELOPED BY**

Catarina Monteiro - 202105279
Gonçalo Monteiro - 202105821
Lara Sousa - 202109782

december 15th 2023

# Overview




- The aim of this project was to  introduce specific changes or customizations to the environment Bipedal Walker, from box2D, and train a reinforcement learning agent using the Stable Baselines library.
- The goal is to assess how these changes impact the agent's learning process and performance.

## The environment chosen’s characteristics

### Action Space

- Actions are motor speed values in the [-1, 1] range for each of the 4 joints at both hips and knees.

## Rewards

- Reward is given for moving forward, totaling 300+ points up to the far end. If the robot falls, it gets -100. Applying motor torque costs a small amount of points. A more optimal agent will get a better score.

## States

- The initial state places the robot standing at the left end with a horizontal hull, legs in a specific position, and a slight knee angle. Episodes end if the hull touches the ground or the robot exceeds the terrain length (200 steps) to the right.

## Percepts (Observations)

- Observations include various details such as hull angle speed, angular velocity, horizontal and vertical speed, joint positions, joint angular speed, leg contact with the ground, and lidar range measurements.

# Changes made to the agent

- Density of the hull;
- Changing the penalties of the actions; (Creation of a penalty for sudden movements - the values were changes to optimize it)

# RL algorithms chosen

- PPO;
- TQC;
- A2C;
- RecurrentPPO;

# Changes made to the environment

This changes were made as a “measure” of the robustness of the model

- Addition of humps;
- Addition of ditches;

# Approach taken

Run the original model with PPO - 10M timesteps

For each change proposed - 3 values were tested with PPO - 500K timesteps

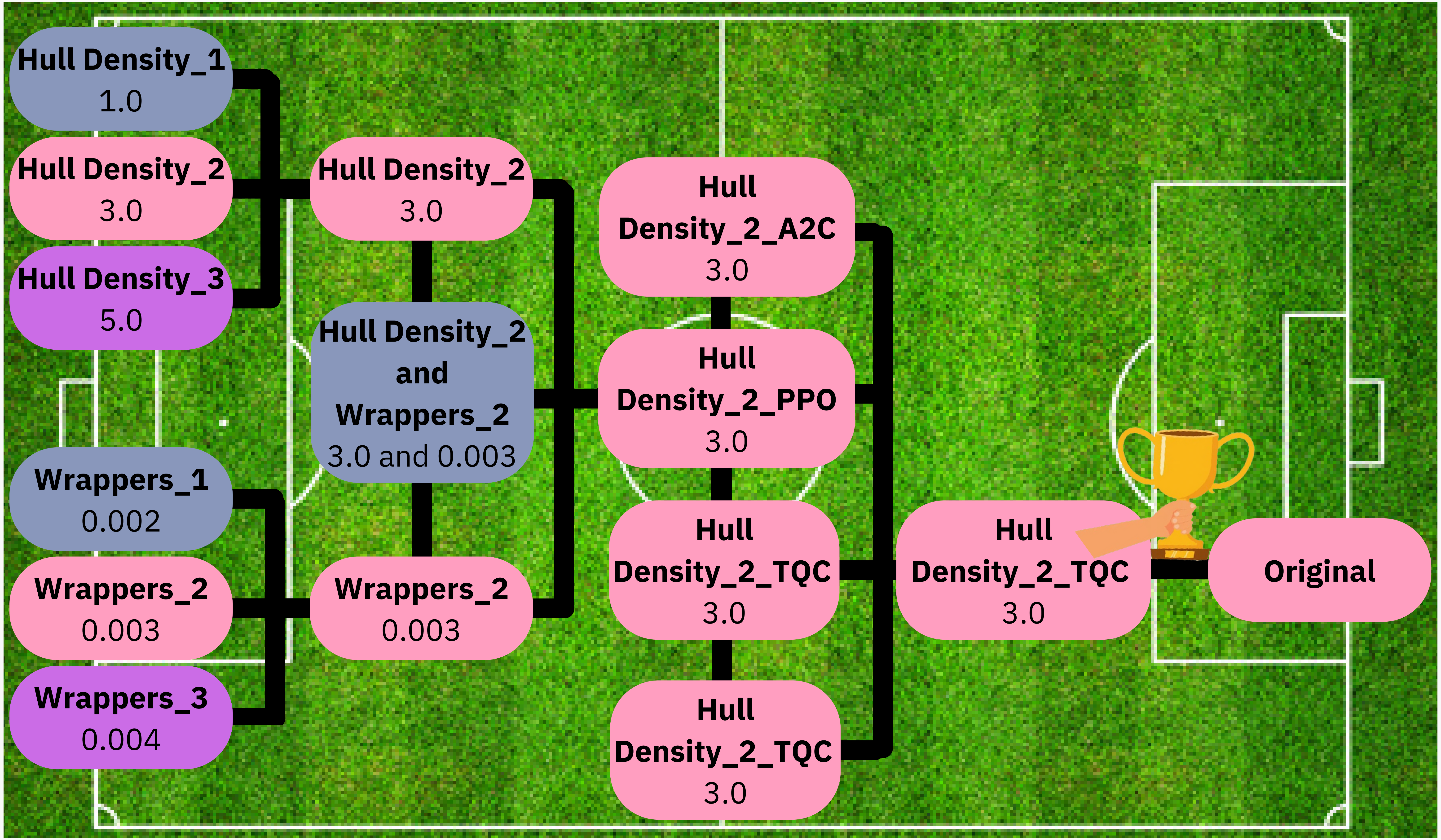
The best modification (for each of the 3) was chosen

Every combination of the best modifications (3) was trained with PPO - 10M timesteps

The one with the best performance was chosen and trained with A2C, RecurrentPPO and TQC - 10M timesteps

Test the robustness of the final model

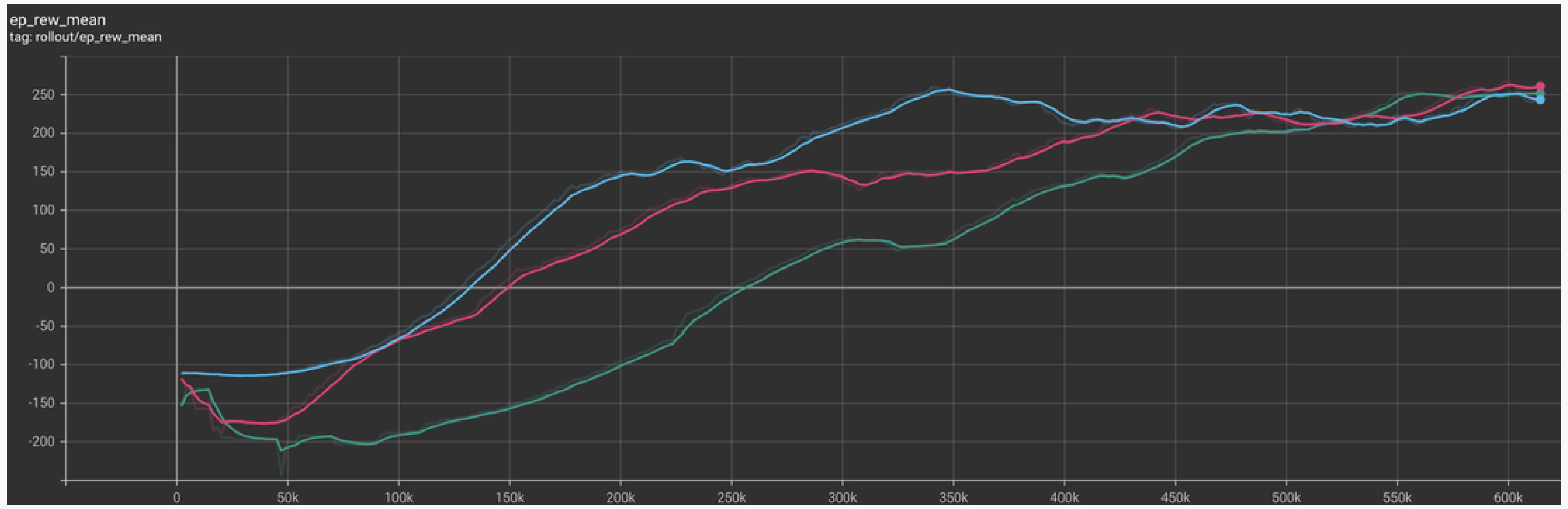
# Experimental Results



# Experimental Results

## Hull changes

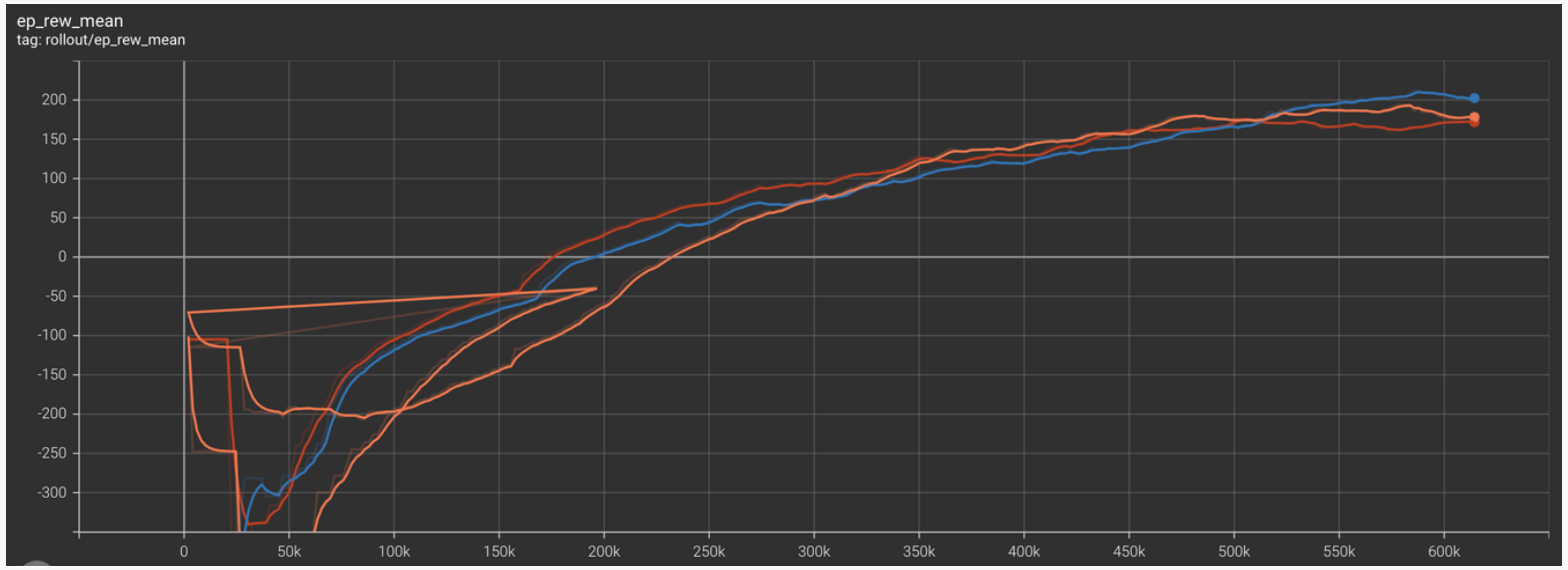
PPO_head_attempt1_0
PPO_head_attempt2_0
PPO_head_attempt3_0





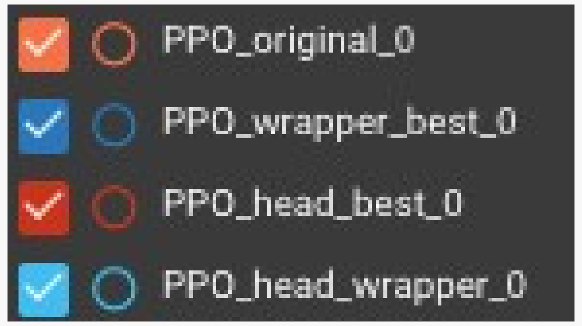
## Penalties changes

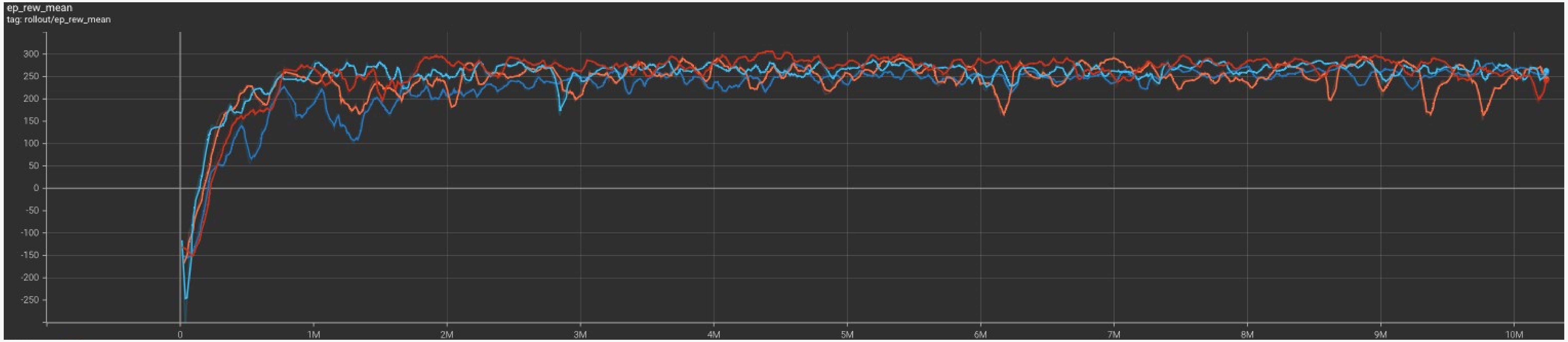
PPO_wrapper_attempt1_0
PPO_wrapper_attempt2_0
PPO_wrapper_attempt3_0

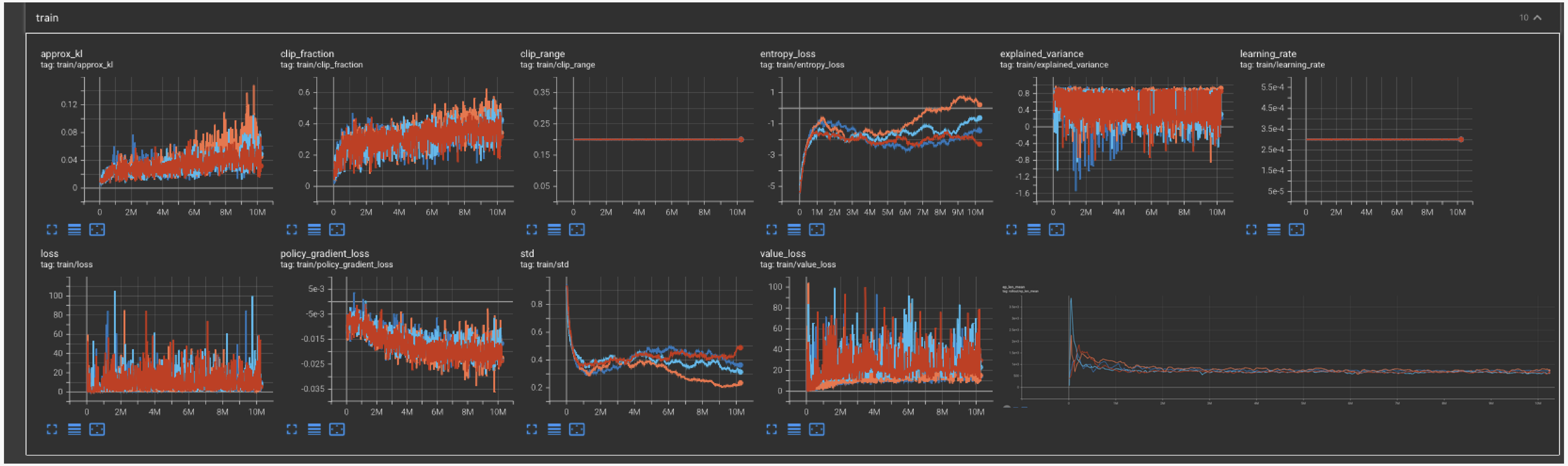



# Experimental Results

## Hull Density_2 - Wrappers_2 - Hull Density_2 and Wrappers_2

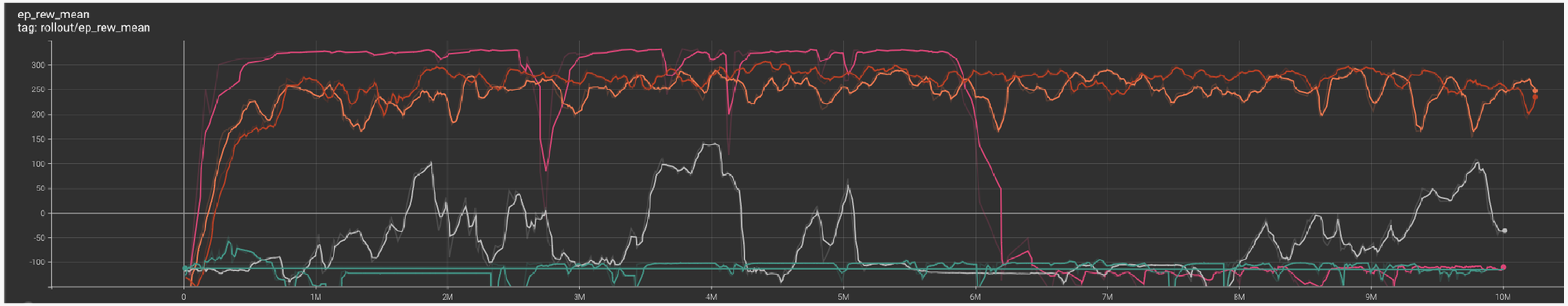


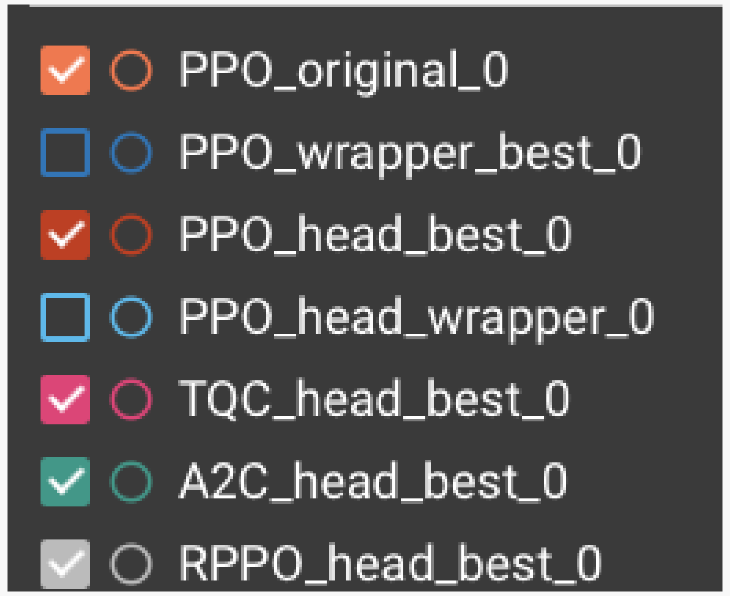



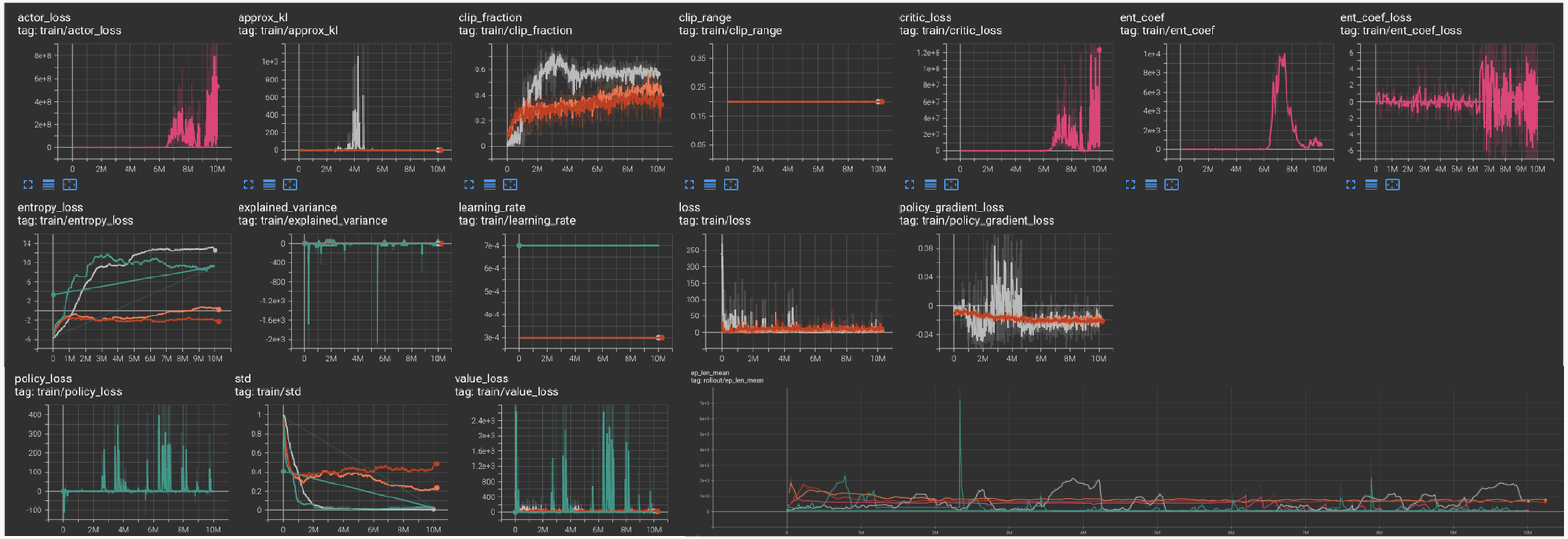



# Experimental Results

## A2C - TQC - PPO - RPPO - Original

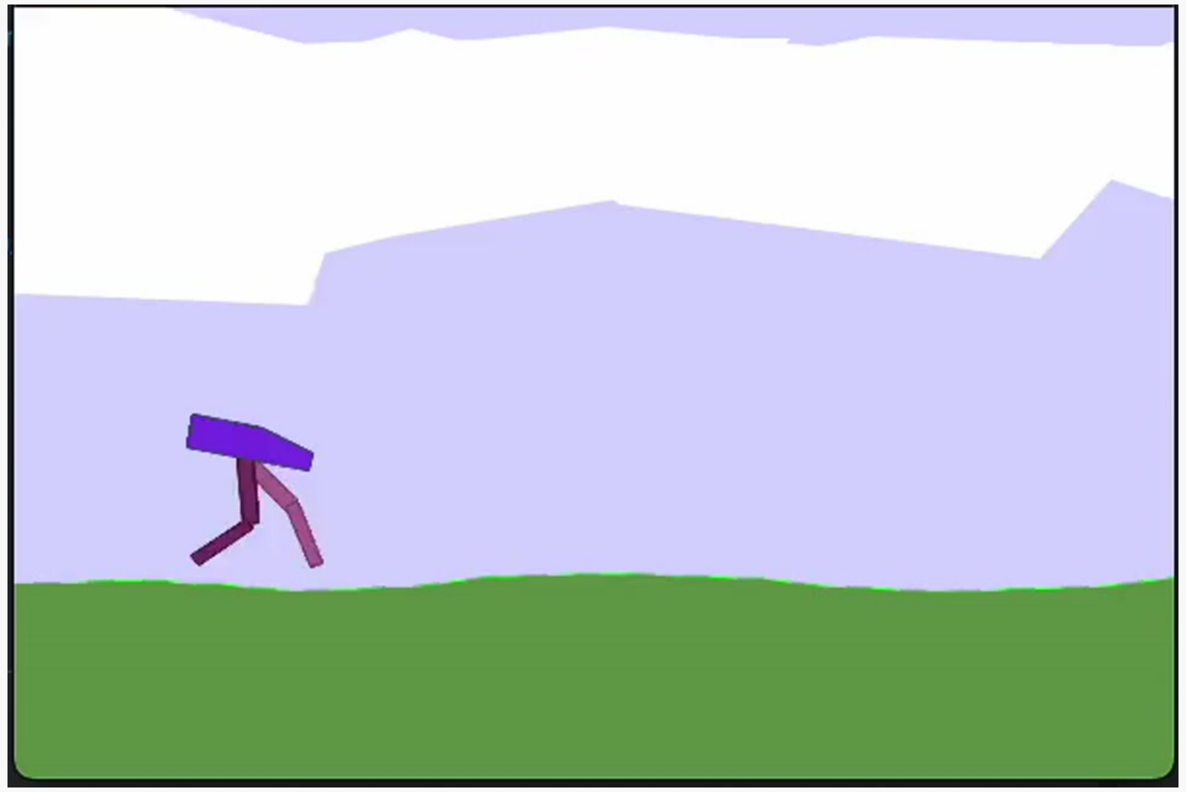


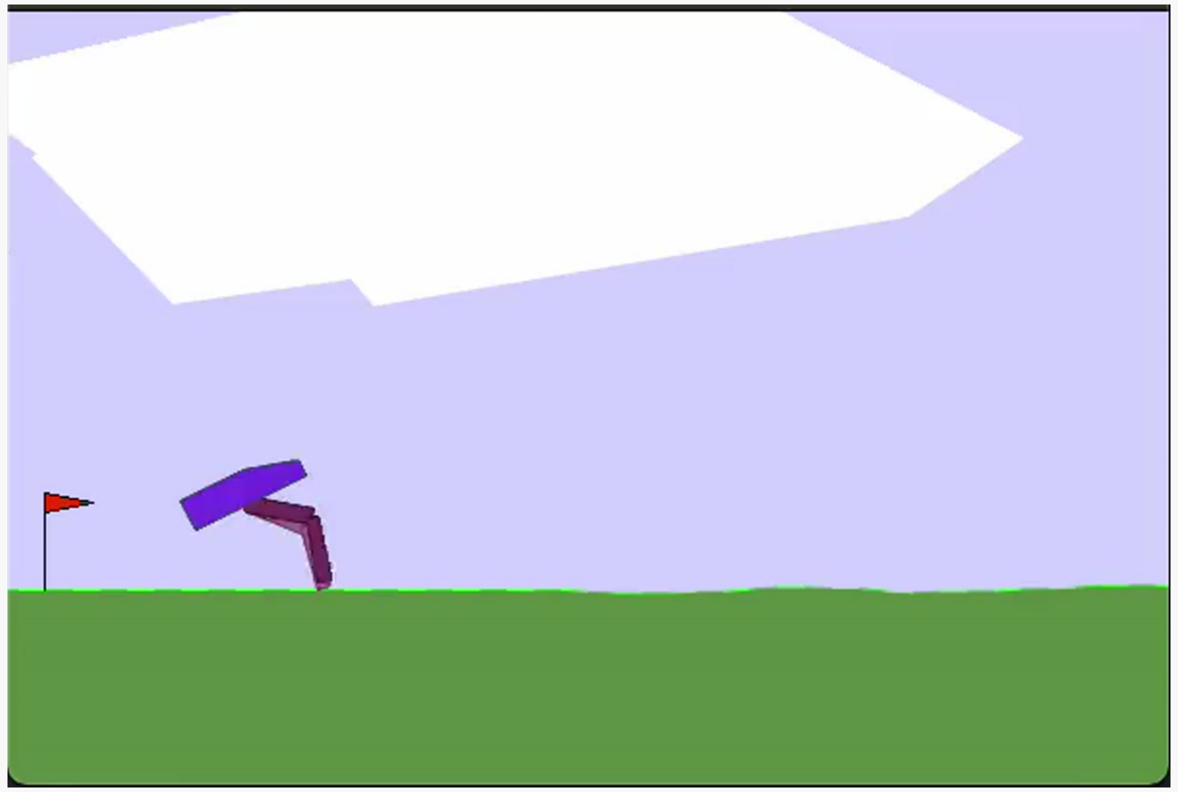



# The best model's performance and robustness test

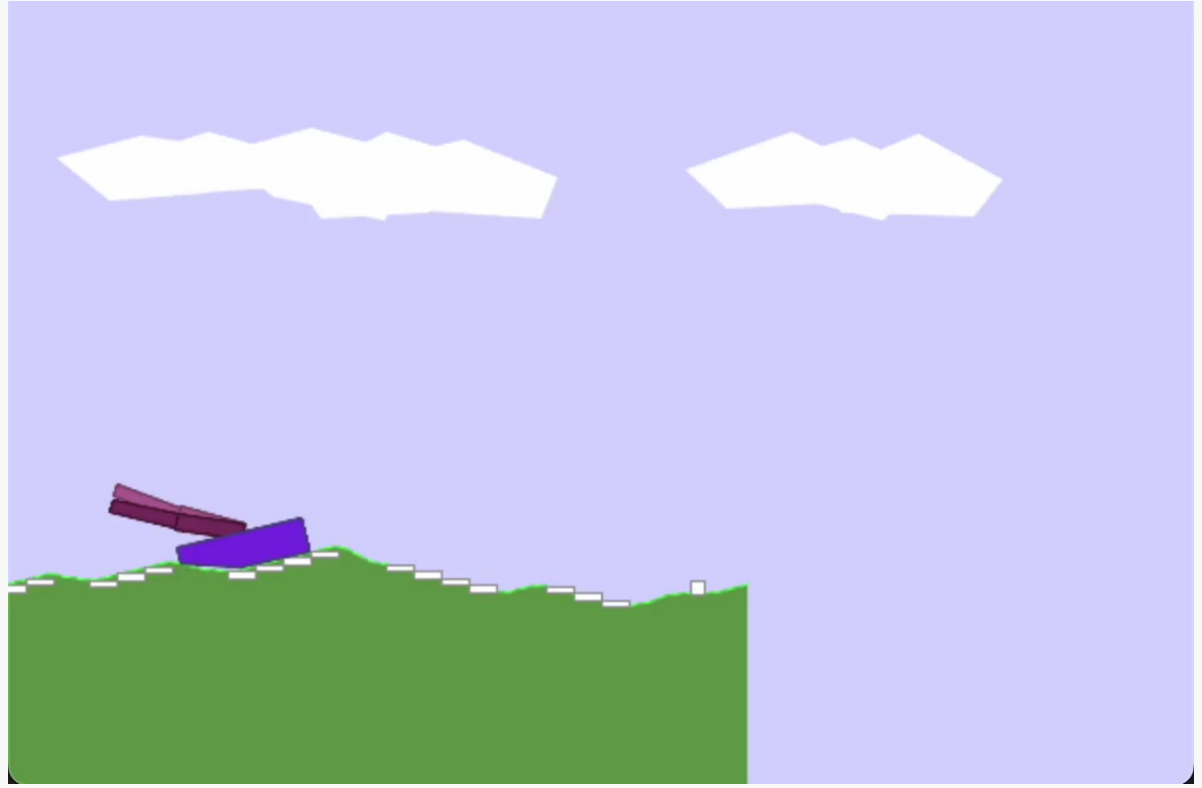
TQC - positive performance



TQC - negative performance



TQC - robustness test



# Conclusions

- It was concluded that the modification that improved the model the most has the decreasing of the hull density.
- As for the algorithms, PPO and TQC were the ones with best performance.
- In sum it was concluded that the best Reinforcement Learning algorithm was TQC, eventhoug it colapsed at 6M.
- TQC reached over 300+ points, which means, it exceeded the maximum reward expected.
- On the future, to obtain a better and more robust model, we should take an similar approach but train the model with more timesteps as well as train the model in a environment with obstacles, so that it is possible for the model to try other exploits