Reinforcement learning project: *Hindsight Goal Ranking for Pick-and-Place tasks*

Master’s Degree in Artificial Intelligence and Robotics

Paradiso Emiliano (1940454)

SAPIENZA
UNIVERSITÀ DI ROMA

Overview

Introduction

- Learning effective policies in sparse reward environments is challenging. This project aims to address the challenges posed by sparse rewards in fixed-base manipulator tasks.

- Simulations were conducted in a Gym environment using the gymnasium-robotics toolkit.

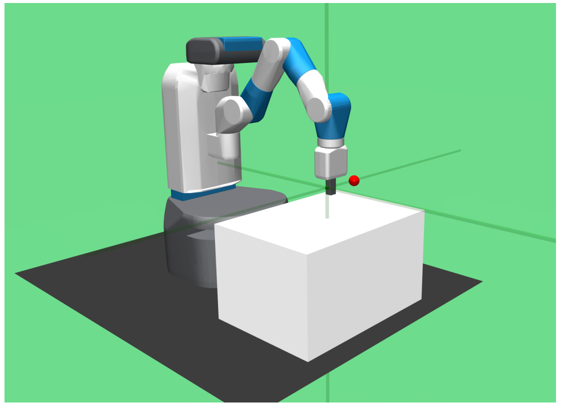
- *Note*: Issues with the default position of the fixed-base manipulator were encountered. For details and troubleshooting, refer to the GitHub repository: [https://github.com/Emilianogith/RL_project-Hindsight-Goal-Ranking-for-Pick-and-Place-tasks/tree/main

- RL algorithm: **DDPG** (Deep Deterministic Policy Gradient).

- **HGR** (Hindsight Goal Ranking) strategy have been used to efficently sample experiences and goals guaranteeeing efficient performance.

Overview
Introduction

- To simplify the problem, we initially trained the agent in a basic environment: gym FetchReach, focusing only on reaching a target position.



- Once the learning strategy was successfully validated in the FetchReach environment, we applied it to the more complex task of gym Fetch PickAndPlace.

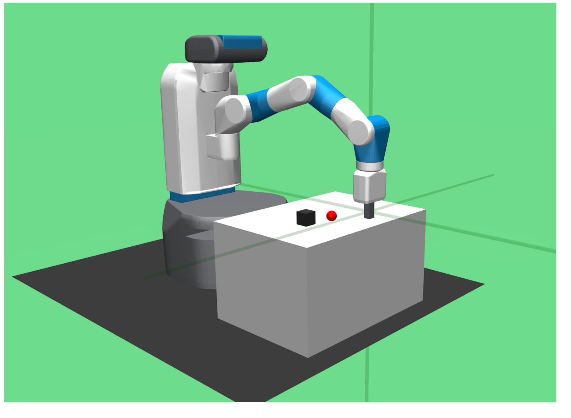


Table of Contents

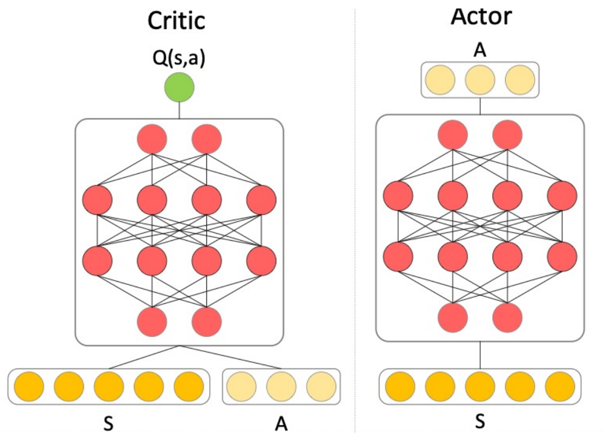
Control

DDPG

The algorithm

- DDPG is a deterministic policy gradient method: specifically adapted for continuous action spaces.

$\mu_\theta\ (s)$: $S \to A$



- Actor-Critic method.

- It optimizes the functions:

Critic : $$L(\phi) = \mathbb{E}_{s\sim\rho^\beta, a\sim\beta, r\sim E, s\sim\rho^\beta}\left[r + \gamma Q_{\phi^-}(s', \mu_{\theta^-}(s')) - Q_\phi(s,a)\right]^2$$

Actor : $$\nabla_\theta J_\beta(\mu_\theta) = \mathbb{E}_{s\sim\rho^\beta}\left[\nabla_\theta \mu_\theta(s) \nabla_a Q_\phi(s,a)|_{a=\mu_\theta(s)}\right]$$

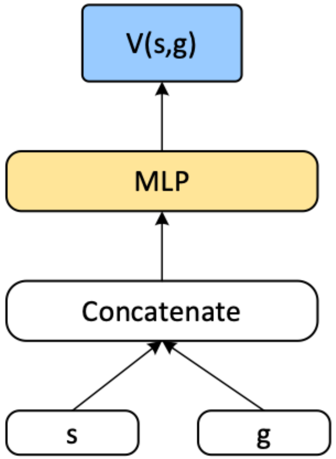
- Off-policy: the target policy $\mu_\theta(s)$ is learned from trajectories generated by an arbitrary stochastic behaviour policy $\pi(s)$.

DDPG

The algorithm

- Target networks, updated using a soft update mechanism, are employed to enhance training stability.
- Gaussian noise with zero mean and a small standard deviation ($\sigma = 0.2$) is added to address exploration challenges associated with the deterministic policy approach.

- In scenarios like these, Critic and actor are goal-conditioned: $\mu_\theta(s|g), Q_\phi(s,a|g)$
- UVFAs are used:



- Hindsight Experience Replay consists in relabeling the goal of the existing experiences in the replay buffer to overcome the sample inefficiency problem in the sparse reward environment.
- Samples are sampled according to the future startegy: the hindsight goal is selected randomly from states at future time steps with respected the chosen experience.

HGR
The algorithm

- In **HER**: the hindsight goal of a sampled episode is selected by uniformly sampling a future visited state.

- In **HGR**: episodes and experieces in the replay buffer are prioritized maximizing what the agent can learn from the whole replay buffer.

  - TD error is used for prioritizing the experiences:
  $$\delta = r + \gamma Q_{\phi^-}(s', \mu_{\theta^-}(s')) - Q_{\phi}(s,a)$$

  - Given a sample episode, the probability of sampling experience j and future goal i with $\mathrm{i} = j+1, \dots, H$:
  $$j, i \sim P'(j,i) = \frac{\left|\delta_{ji}\right|^{\alpha'}}{\sum_{j=1}^{H-1} \sum_{i=j+1}^{H} \left|\delta_{ji}\right|^{\alpha'}}$$

  - Also the episodes are prioritized accordingly to the average TD error($\delta^{(n)}$) of all experiences within the episodes:
  $$n \sim P(n) = \frac{\left|\delta^{(n)}\right|^{\alpha}}{\sum_n \left|\delta^{(n)}\right|^{\alpha}}$$

Table of Contents

Trajectories

Description

FetchReach

- **State space:**

0	1	2	3	4	5	6	7	8	9
Ee_x	Ee_y	Ee_z	Grip_r	Grip_l	V_x	V_y	V_z	Grip_vl	Grip_vr

- **Action space:**

0	Displacement of the end effector in the x direction dx
1	Displacement of the end effector in the y direction dy
2	Displacement of the end effector in the z direction dz
3	-

- The task in Gym FetchReach is for the agent to move the robot's end-effector to a specified target position in space.

Implementation details

FetchReach

- The reward is sparse and it is defined as:

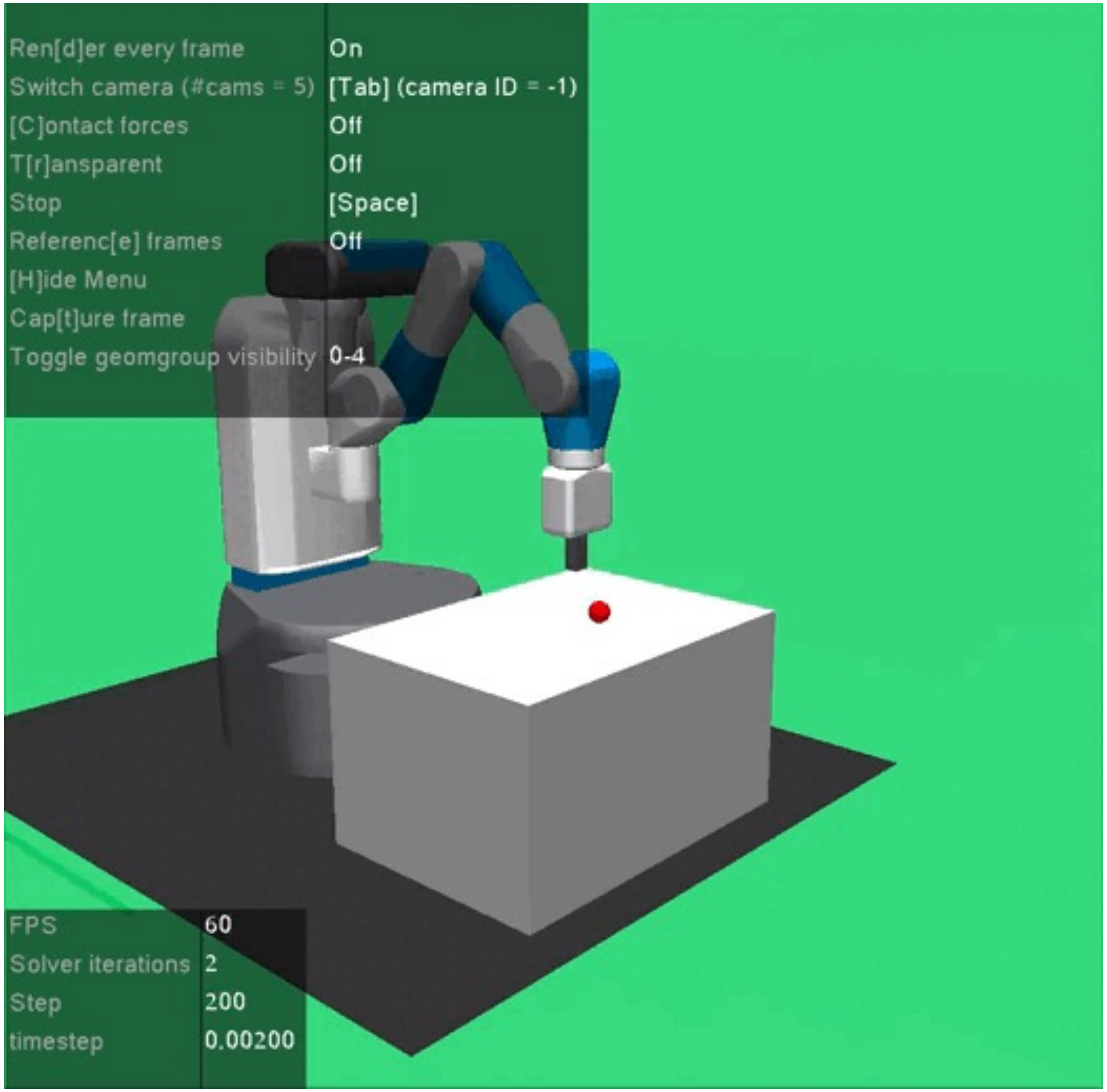
$$r = \begin{cases} 0 & if\ |ee - goal| < \tau \\ -1 & otherwise \end{cases}$$

- Gaussian noise alone is insufficient to ensure effective exploration. To address this, we enhanced exploration by incorporating the epsilon-greedy strategy.

$$a = \begin{cases} random & if\ \epsilon \\ \beta(s) & if\ 1 - \epsilon \end{cases}$$

Evaluation

FetchReach



Evaluation

FetchReach

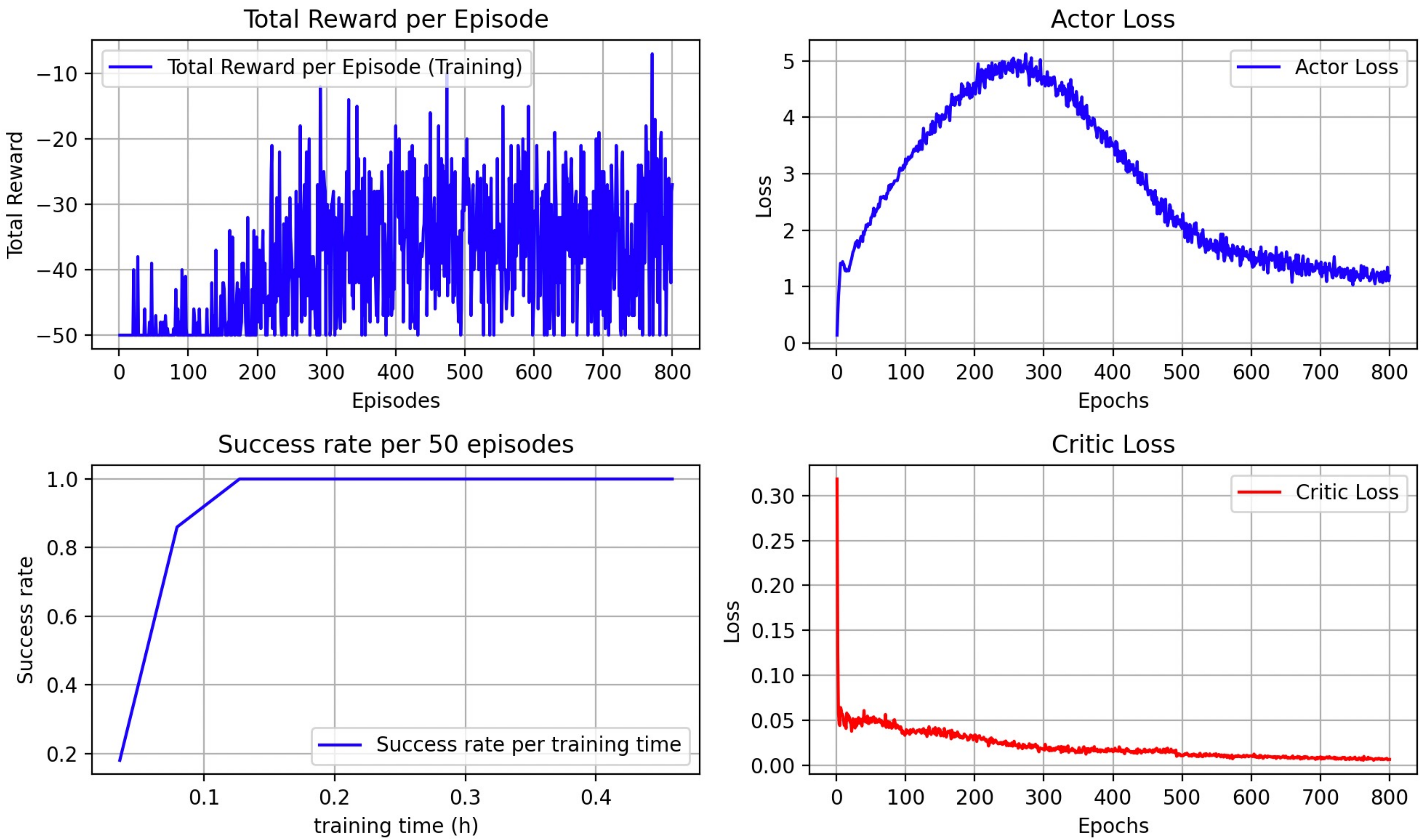


Table of Contents

Trajectories

Description
FetchPickAndPlace

- **State space:**

0	1	2	3	4	5	6	7	8	9
Ee_x	Ee_y	Ee_z	Block_ x	Block_ y	Block_ z	X_b- x_g	Y_b- y_g	Z_b- z_g	Grip_r
10	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
Grip_l	Block θx	Block θy	Block θz	Vx_b- vx_g	Vy_b- vy_g	Vz_b- vz_g	Block ω_x	Block ω_y	Block ω_z
20	**21**	**22**	**23**	**24**					
Ee_Vx	Ee_Vy	Ee_Vz	Grip_Vr	Grip_Vl					

- **Action space:**

0	Displacement of the end effector in the x direction dx
1	Displacement of the end effector in the y direction dy
2	Displacement of the end effector in the z direction dz
3	Positional displacement per timestep of each finger of the gripper

- In Gym FetchPickAndPlace, the agent must not only reach a target position but also manipulate an object to a desired location.

Implementation details

FetchPickAndPlace

- Here the task is more articulated, we need to redefine the reward function for hindsight goal as:

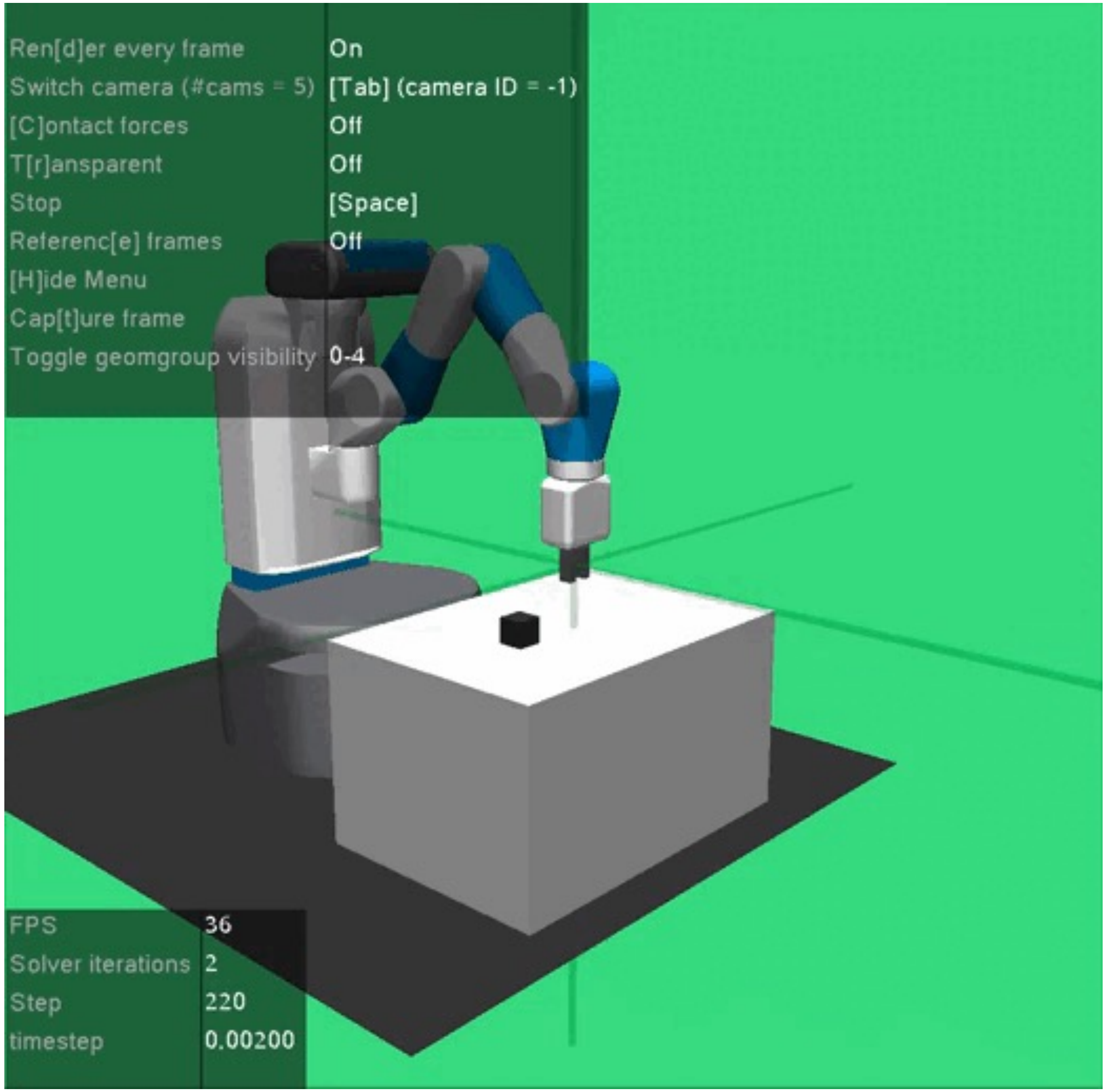
$$r = \begin{cases} 0 & \text{if } |\text{block}-\text{goal}| < \tau \text{ and } \text{block} \neq \text{goal} \\ -1 & \textit{otherwise} \end{cases}$$

- Gaussian noise alone is insufficient to ensure effective exploration. To address this, we enhanced exploration by incorporating the epsilon-greedy strategy.

$$a = \begin{cases} random & if\ \epsilon \\ \beta(s) & if\ 1-\epsilon \end{cases}$$

Evaluation

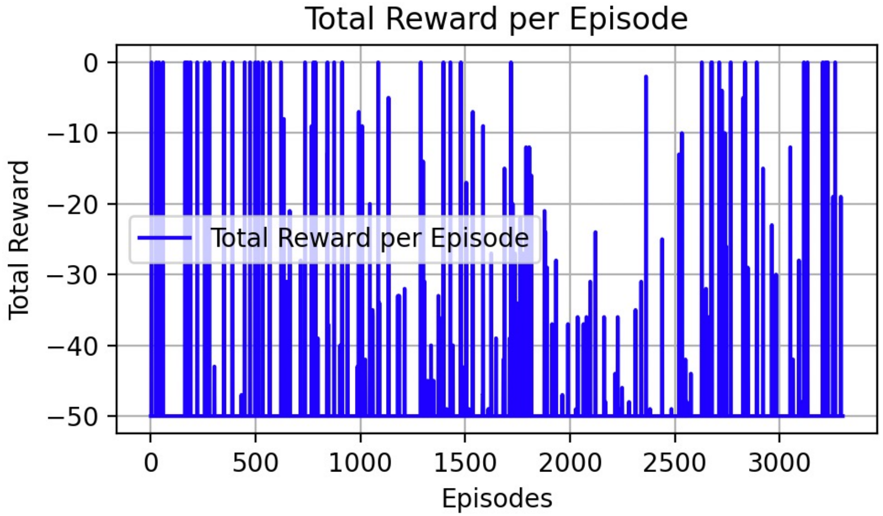
FetchPickAndPlace

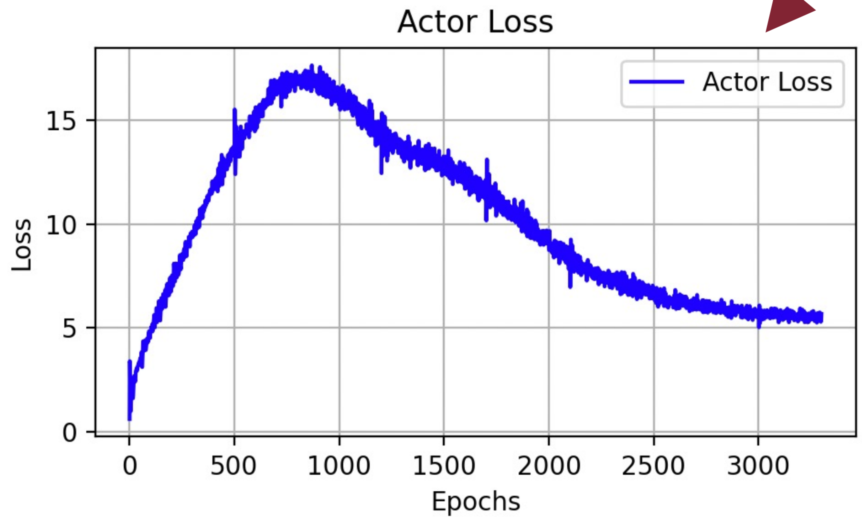


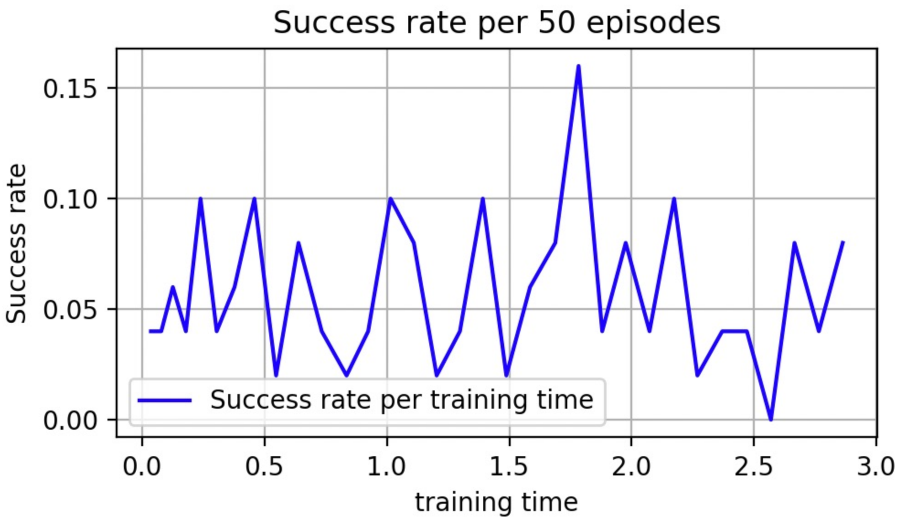
Evaluation

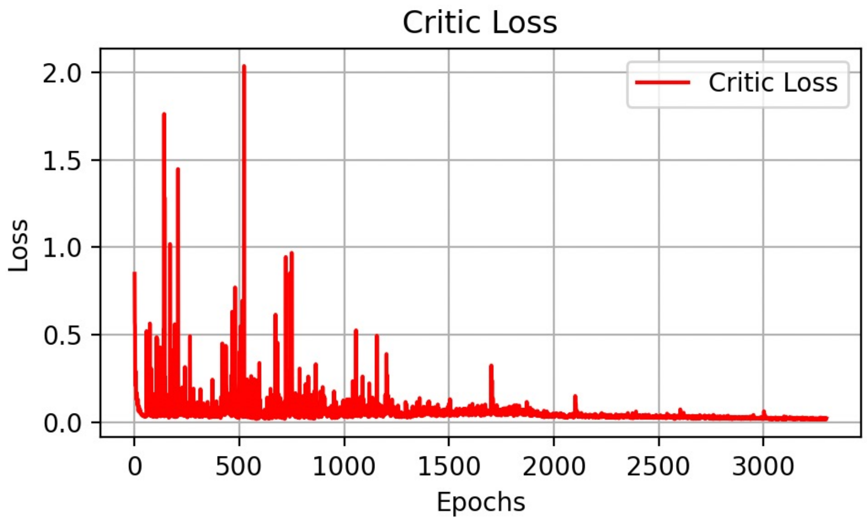
FetchPickAndPlace

Trained only for 3300 epochs









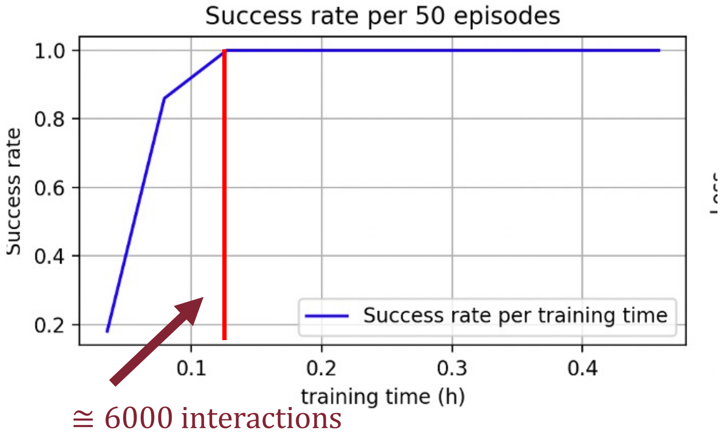
Conclusions

Conclusions

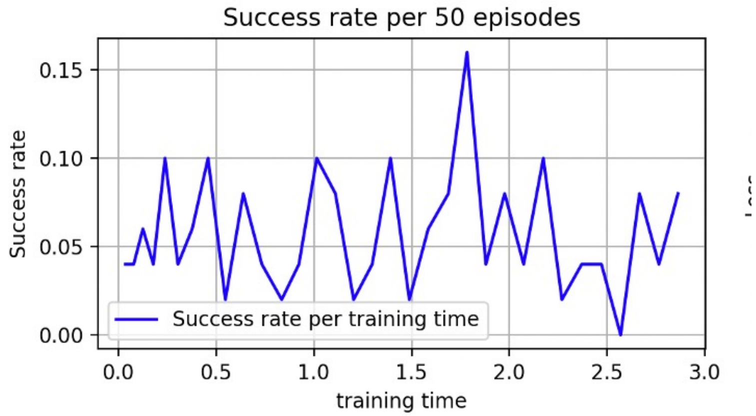
- **DDPG** demonstrated strong suitability for learning in continuous action spaces and solving multi-goal tasks.

- **HGR** showed effective sampling efficiency, as shown by higher win rates, correctly addressing the problem of sparse reward.

In Gym FetchReach:



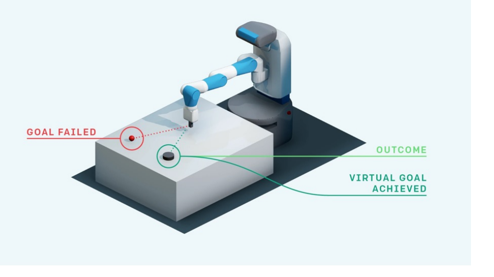
In Gym FetchPickAndPlace not very effective....

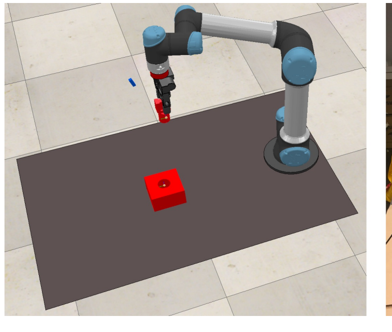


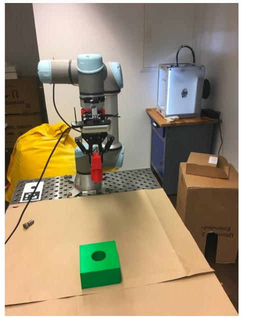
Future Works

Conclusions

- **Opportunities for growth:**
    - Try different exploration strategies
    - Train for more epochs
    - Leverage deeper networks
    - Reshape the reward
- **Future works:**
    - Expand the approach to different tasks, including more complex and dynamic scenarios.
    - Validate the method on a physical manipulator to address Sim-to-Real transfer challenges.







Thanks for your attention!