

Search

INTRODUCTION

Installation

Multi-goal API

ENVIRONMENTS

Fetch

Pick And Place

Push

Reach

Slide

Shadow Dexterous Hand

Maze

Adroit Hand

Franka Kitchen

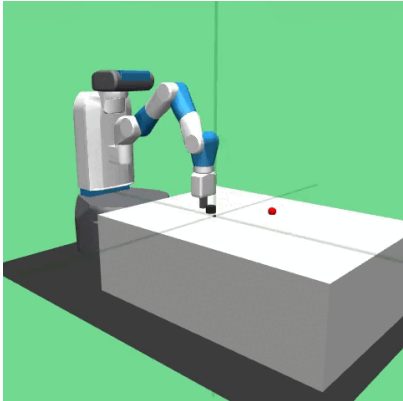
MaMuJoCo (Multi-Agent MuJoCo)

DEVELOPMENT

Github

Release Notes

# Slide



## Description

This environment was introduced in ["Multi-Goal Reinforcement Learning: Challenging Robotics Environments and Request for Research"](#).

The task in the environment is for a manipulator hit a puck in order to reach a target position on top of a long and slippery table. The table has a low friction coefficient in order to make it slippery for the puck to slide and be able to reach the target position which is outside of the robot's workspace. The robot is a 7-DoF [Fetch Mobile Manipulator](#) with a two-fingered parallel gripper. The robot is controlled by small displacements of the gripper in Cartesian coordinates and the inverse kinematics are computed internally by the MuJoCo framework. The gripper is locked in a closed configuration since the puck doesn't need to be grasped. The task is also continuing which means that the robot has to maintain the puck in the target position for an indefinite period of time.

The control frequency of the robot is of `f = 25 Hz`. This is achieved by applying the same action in 20 subsequent simulator step (with a time step of `dt = 0.002 s`) before returning the control to the robot.

## Action Space

The action space is a `Box(-1.0, 1.0, (4, ), float32)`. An action represents the Cartesian displacement dx, dy, and dz of the end effector. In addition to a last action that controls closing and opening of the gripper.

Num	Action	Control Min	Control Max	Name (in corresponding XML file)	Joint	Unit
0	Displacement of the end effector in the x direction dx	-1	1	robot0:mocap	hinge	position (m)
1	Displacement of the end effector in the y direction dy	-1	1	robot0:mocap	hinge	position (m)
2	Displacement of the end effector in the z direction dz	-1	1	robot0:mocap	hinge	position (m)
3	-	-1	1	-	-	-

## Observation Space

The observation is a `goal-aware observation space`. It consists of a dictionary with information about the robot's end effector state and goal. The kinematics observations are derived from Mujoco bodies known as [sites](#) attached to the body of interest such as the puck or the end effector. Only the observations from the gripper fingers are derived from joints. Also to take into account the temporal influence of the step time, velocity values are multiplied by the step time `dt=number_of_sub_steps*sub_step_time`. The dictionary consists of the following 3 keys:

- `observation`: its value is an `ndarray` of shape `(25, )`. It consists of kinematic information of the puck object and gripper. The elements of the array correspond to the following:

Num	Observation	Min	Max	Site Name (in corresponding XML file)	Joint Name (in corresponding XML file)	Joint Type	Unit
0	End effector x position in global coordinates	-Inf	Inf	robot0:grip	-	-	positic (m)
1	End effector y position in global coordinates	-Inf	Inf	robot0:grip	-	-	positic (m)
2	End effector z position in global coordinates	-Inf	Inf	robot0:grip	-	-	positic (m)
3	Puck x position in global coordinates	-Inf	Inf	object0	-	-	positic (m)
4	Puck y position in global coordinates	-Inf	Inf	object0	-	-	positic (m)
5	Puck z position in global coordinates	-Inf	Inf	object0	-	-	positic (m)
6	Relative puck x position with respect to gripper x position in globla coordinates. Equals to $X_{gripper} - X_{puck}$	-Inf	Inf	object0	-	-	positic (m)
7	Relative puck y position with respect to gripper y position in globla coordinates. Equals to $Y_{gripper} - Y_{puck}$	-Inf	Inf	object0	-	-	positic (m)
8	Relative puck z position with respect to gripper z position in globla coordinates. Equals to $Z_{gripper} - Z_{puck}$	-Inf	Inf	object0	-	-	positic (m)
9	Joint displacement of the right gripper finger	-Inf	Inf	-	robot0:r_gripper_finger_joint	hinge	positic (m)
10	Joint displacement of the left gripper finger	-Inf	Inf	-	robot0:l_gripper_finger_joint	hinge	positic (m)
11	Global x rotation of the puck in a XYZ Euler frame rotation	-Inf	Inf	object0	-	-	angle (rad)
12	Global y rotation of the puck in a XYZ Euler frame rotation	-Inf	Inf	object0	-	-	angle (rad)
13	Global z rotation of the puck in a XYZ Euler frame	-Inf	Inf	object0	-	-	angle (rad)

Num	Observation	Min	Max	Site Name (in corresponding XML file)	Joint Name (in corresponding XML file)	Joint Type	Unit
14	Relative puck linear velocity in x direction with respect to the gripper	-Inf	Inf	object0	-	-	velocity (m/s)
15	Relative puck linear velocity in y direction with respect to the gripper	-Inf	Inf	object0	-	-	velocity (m/s)
16	Relative puck linear velocity in z direction	-Inf	Inf	object0	-	-	velocity (m/s)
17	Puck angular velocity along the x axis	-Inf	Inf	object0	-	-	angular velocity (rad/s)
18	Puck angular velocity along the y axis	-Inf	Inf	object0	-	-	angular velocity (rad/s)
19	Puck angular velocity along the z axis	-Inf	Inf	object0	-	-	angular velocity (rad/s)
20	End effector linear velocity x direction	-Inf	Inf	robot0:grip	-	-	velocity (m/s)
21	End effector linear velocity y direction	-Inf	Inf	robot0:grip	-	-	velocity (m/s)
22	End effector linear velocity z direction	-Inf	Inf	robot0:grip	-	-	velocity (m/s)
23	Right gripper finger linear velocity	-Inf	Inf	-	robot0:r_gripper_finger_joint	hinge	velocity (m/s)
24	Left gripper finger linear velocity	-Inf	Inf	-	robot0:l_gripper_finger_joint	hinge	velocity (m/s)

- `desired_goal`: this key represents the final goal to be achieved. In this environment it is a 3-dimensional `ndarray`, `(3,)`, that consists of the three cartesian coordinates of the desired final puck position `[x,y,z]`. In order for the robot to perform a pick and place trajectory, the goal position can be elevated over the table or on top of the table. The elements of the array are the following:

Num	Observation	Min	Max	Site Name (in corresponding XML file)	Unit
0	Final goal puck position in the x coordinate	-Inf	Inf	target0	position (m)
1	Final goal puck position in the y coordinate	-Inf	Inf	target0	position (m)
2	Final goal puck position in the z coordinate	-Inf	Inf	target0	position (m)

- `achieved_goal`: this key represents the current state of the puck, as if it would have achieved a goal. This is useful for goal orientated learning algorithms such as those that use [Hindsight Experience Replay](#) (HER). The value is an `ndarray` with shape `(3,)`. The elements of the array are the following:

Num	Observation	Min	Max	Site Name (in corresponding XML file)	Unit
0	Current puck position in the x coordinate	-Inf	Inf	object0	position (m)
1	Current puck position in the y coordinate	-Inf	Inf	object0	position (m)
2	Current puck position in the z coordinate	-Inf	Inf	object0	position (m)

## Rewards

- *sparse*: the returned reward can have two values: `-1` if the puck hasn't reached its final target position, and `0` if the puck is in the final target position (the puck is considered to have reached the goal if the Euclidean distance between both is lower than 0.05 m).
- *dense*: the returned reward is the negative Euclidean distance between the achieved goal position and the desired goal.

To initialize this environment with one of the mentioned reward functions the type of reward must be specified in the id string when the environment is initialized. For `sparse` reward the id is the default of the environment, `FetchSlide-v2`. However, for `dense` reward the id must be modified to `FetchSlideDense-v2` and initialized as follows:

```
import gymnasium as gym

env = gym.make('FetchSlideDense-v2')
```

## Starting State

When the environment is reset the gripper is placed in the following global cartesian coordinates  $(x, y, z) = [1 \ 0.75 \ 0.41] \text{ m}$ , and its orientation in quaternions is  $(w, x, y, z) = [1.0, \ 0.0, \ 1.0, \ 0.0]$ . The joint positions are computed by inverse kinematics internally by MuJoCo. The base of the robot will always be fixed at  $(x, y, z) = [0.405, \ 0.48, \ 0]$  in global coordinates.

The puck's position has a fixed height of  $(z) = [0.42] \text{ m}$  (on top of the table). The initial  $(x, y)$  position of the puck is the gripper's x and y coordinates plus an offset sampled from a uniform distribution with a range of  $[-0.1, \ 0.1] \text{ m}$ . Offset samples are generated until the 2-dimensional Euclidean distance from the gripper to the puck is greater than  $0.1 \text{ m}$ . The initial orientation of the puck is the same as for the gripper,  $(w, x, y, z) = [1.0, \ 0.0, \ 1.0, \ 0.0]$ .

Finally the target position where the robot has to move the puck is generated. The target can be in mid-air or over the table. The random target is also generated by adding an offset to the initial grippers position  $(x, y)$  sampled from a uniform distribution with a range of  $[-0.3, \ 0.3] \text{ m}$ . The height of the target is initialized at  $(z) = [0.42] \text{ m}$ .

## Episode End

The episode will be `truncated` when the duration reaches a total of `max_episode_steps` which by default is set to 50 timesteps. The episode is never `terminated` since the task is continuing with infinite horizon.

## Arguments

To increase/decrease the maximum number of timesteps before the episode is `truncated` the `max_episode_steps` argument can be set at initialization. The default value is 50. For example, to increase the total number of timesteps to 100 make the environment as follows:

```
import gymnasium as gym

env = gym.make('FetchSlide-v2', max_episode_steps=100)
```

## Version History

- v2: the environment depends on the newest [mujoco python bindings](#) maintained by the MuJoCo team in Deepmind.
- v1: the environment depends on `mujoco_py` which is no longer maintained.

