21st Workshop **From Objects to Agents** (WOA 2020)

RL for Autonomous Agents Exploring Environments: an Experimental Framework and Preliminary Results

Nassim Habbash, Federico Bottoni, Giuseppe Vizzari

RL Applications

- Sea and land rescue drones
- Drone deliveries
- Territorial surveys
- Simulation of complex systems

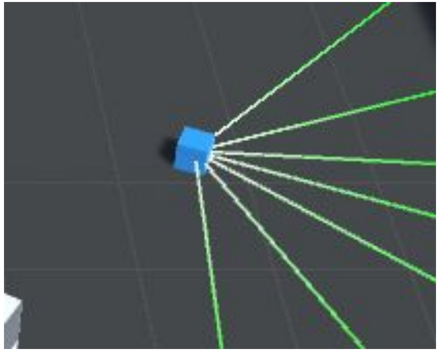
Steps

Definition of a **single-agent obstacle-ridden**, **randomly generated** environment

Training of the agent with Proximal Policy Optimization for a **target localization objective**

Evaluation and comparison of the agent's **model variations**

Perception and action model



Observation space

Tuple of n LIDAR readings at timestep t

$$s = \{(x_1, o_1), (x_2, o_2), ..., (x_n, o_n)\}, s \in S$$

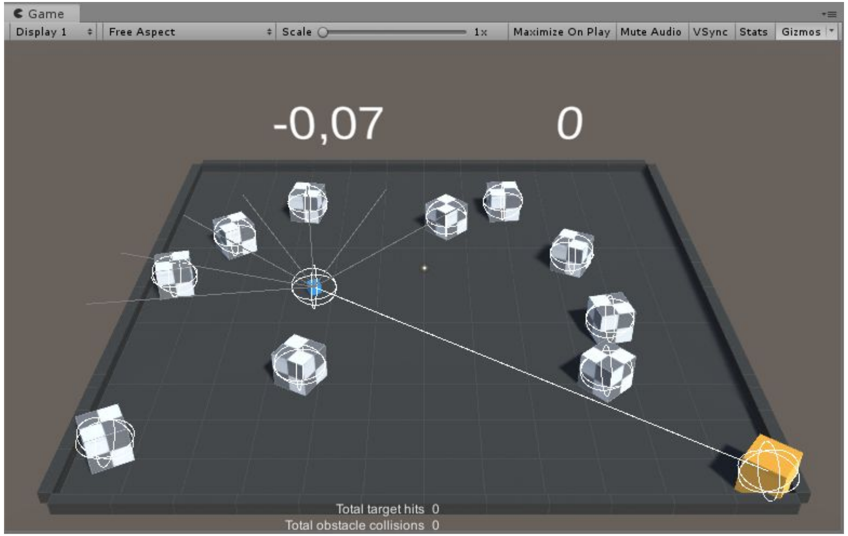
Action space

Possible movements available to the agent

$$A = \{Forward, Side, Rotation\}$$

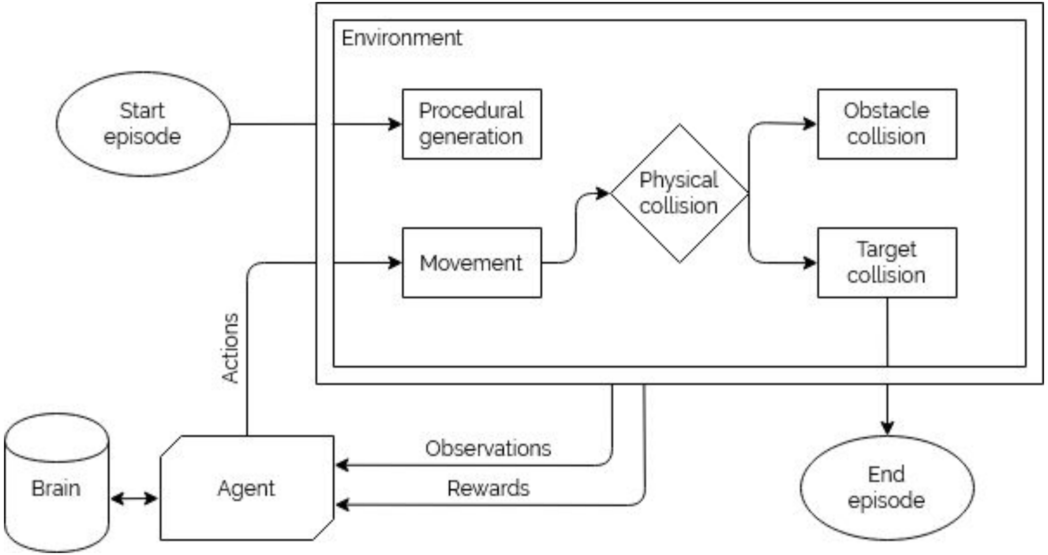
Environment model

- Agent follows **simplified** rover physics
- Movement happens through **force application**
- **Instant velocity change** to the agent's body
- **Soft clamping** to a max velocity
- **Parametrized environment generation** (obstacle density, agent-target distance, etc.)

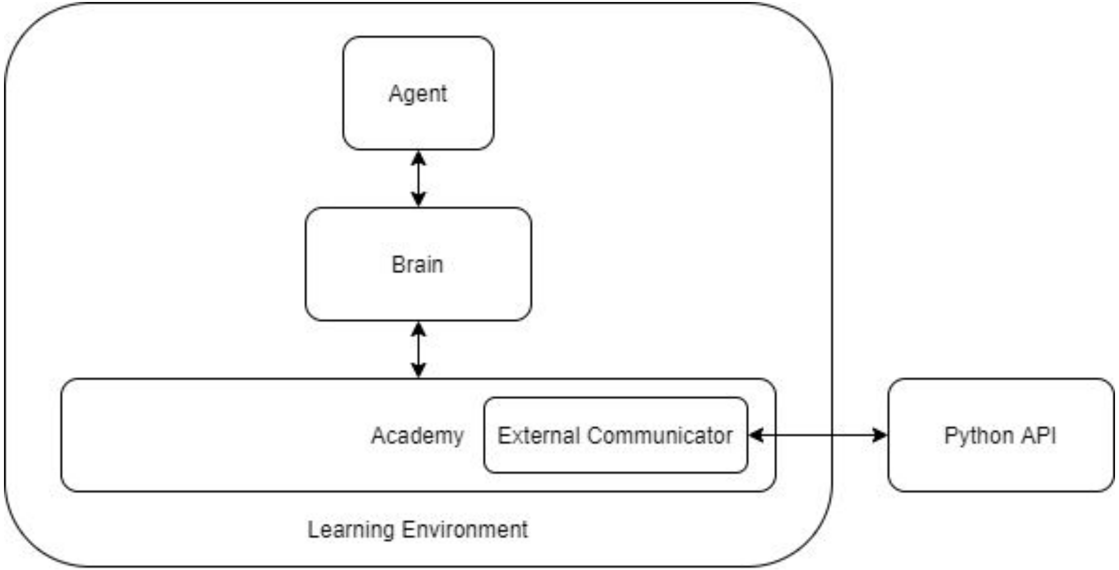


System Architecture

Schema illustrating the **interactions** between the main **actors** of the system



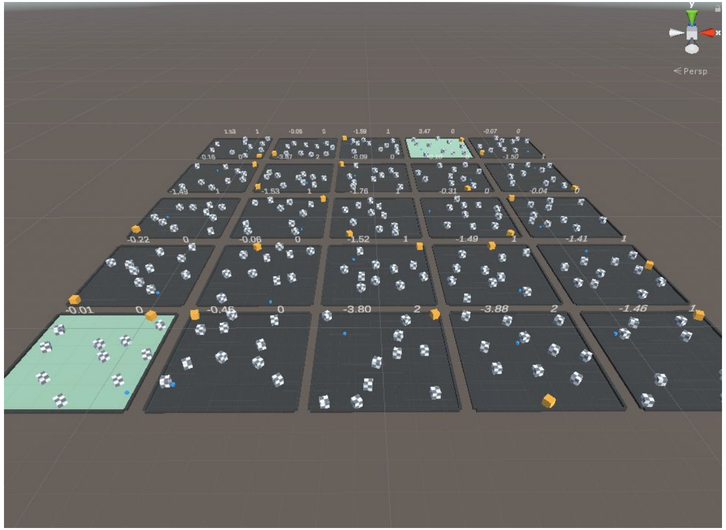
System Architecture



Simulation environment to learning system communication pipeline

Learning System

- Uses **Proximal Policy Optimization** as a RL algorithm (Policy Gradient)
- The **reward signal** defines the goal of the task
- **Curriculum learning** scales the difficulty of the task according to the **cumulative reward** reached by the agent
- The same agent's **Brain** is trained on **parallel environments**



Reward Signal - Extrinsic Reward

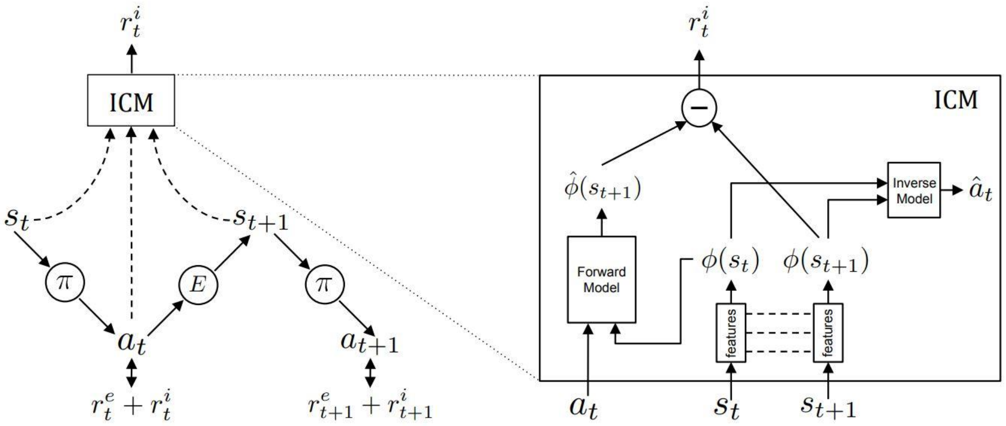
Given in **response** to the **actions** made by the agent in the environment, comprised of two components:

- Reward: *Positive reward - Penalty function*
  - Penalty function: α*obstacle_collisions+β*time*
  - Positive reward: γ*target_collisions*

Reward Signal - Intrinsic Reward

Representing the **curiosity** of the agent.

The more **unexpected** the action taken by the agent, the **higher** the curiosity signal.



Alexei A. Efros Trevor Darrell Deepak Pathak, Pulkit Agrawal. *"Curiosity-driven exploration by self-supervised prediction" 2017.*

Curriculum Learning

Dynamically change the parameters **during training**, making the task **progressively** harder (through increasing lessons).

In some cases, allows for **faster policy convergence.**

Curriculum parameters
Reward threshold
Number of obstacles
Minimum object spawn distance
Target-Agent distance
Penalty offset

Experiments - Training Performance

Baseline

- Single lesson
- Standard penalty func.
- LIDARs

Curriculum

- Multi lessons
- Standard penalty func.
- LIDARs

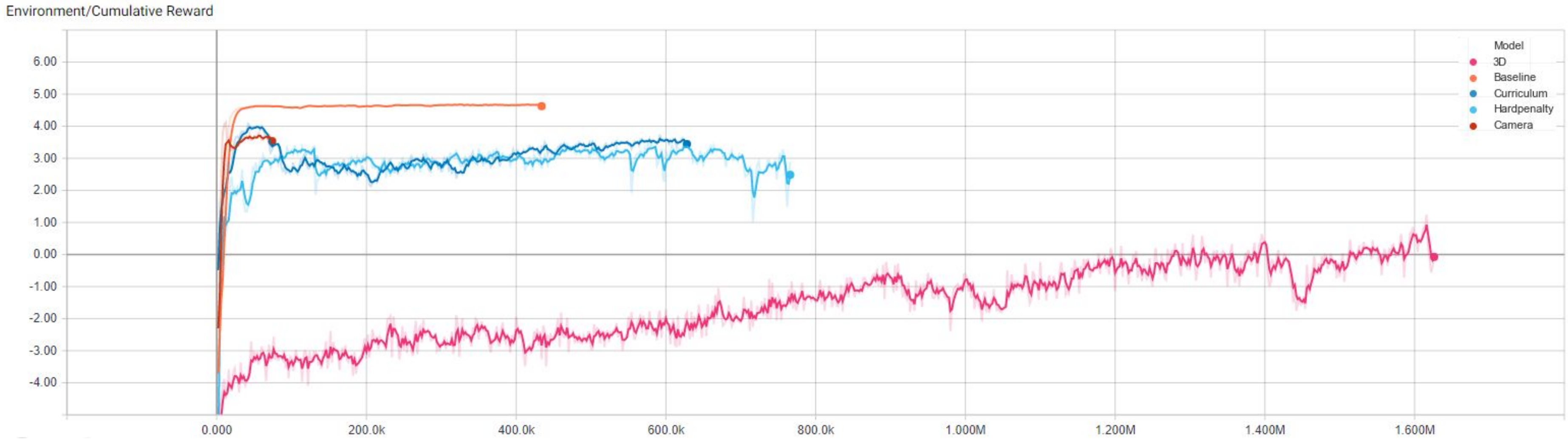
Harder penalty

- Multi lessons
- Harder penalty func.
- LIDARs

Camera

- Multi lessons
- Standard penalty func.
- CAMERA

Results - Performance measures



Results - Performance measures

The experiments converge at different cumulative rewards and reach different curriculum lessons

So, which is the best model? The one with a better reward convergence? The one that advanced to the last lesson?

Which measures are significative of the model performance in the experimental scenario?

Experiments - Evaluation

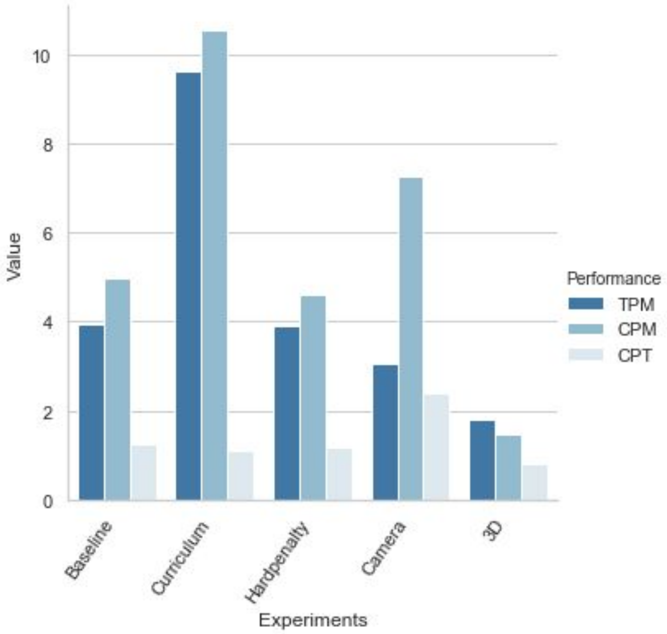
Standard Learning metrics

- Cumulative reward
- Extrinsic and Intrinsic reward
- Policy loss
- Episode Length
- ...

Environmental - domain specific - Performance metrics

- Collisions per minute (CPM)
- Targets per minute (TPM)
- Collisions per target (CPT)

Results - Training performance

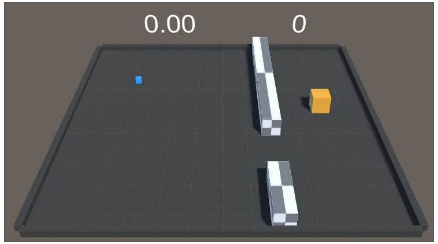


Best overall: ***Curriculum Learning Model.***

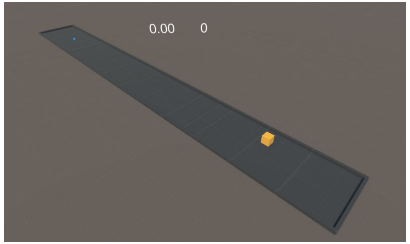
- *Curriculum model* behaves as accurately as *Baseline* but it is quicker
- Almost every model is **below 2 CPT**
- The *harder penalty model* **didn't bring any improvement** to the performance
- The *camera model* **did not perform** as well as the LIDAR model...
    - ... but the learning process was short, and training more could change the situation

Experiments - Exp. Transferability

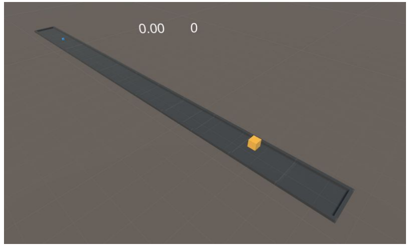
Rooms



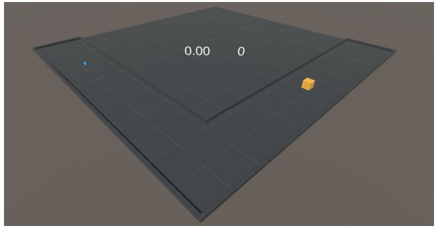
Corridor



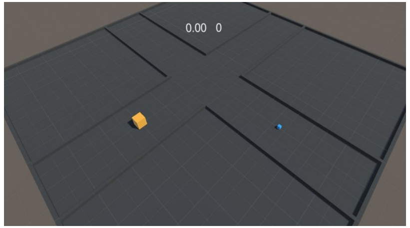
Corridor Tight



Turn

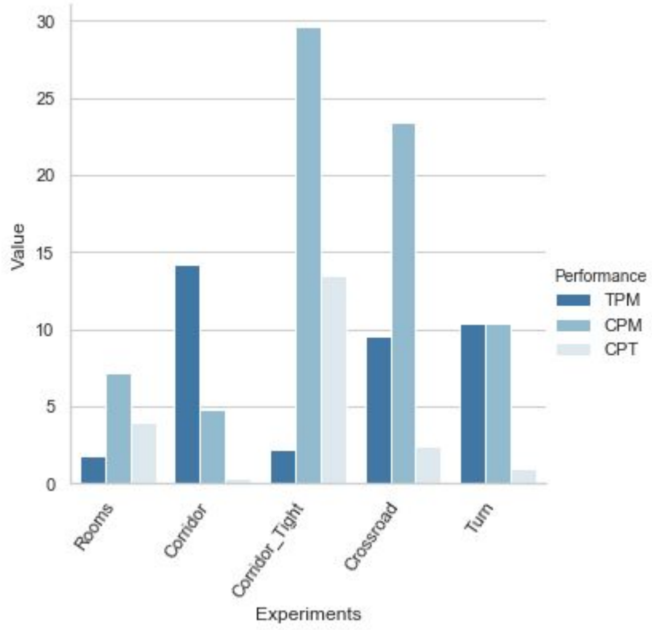


Crossroad



Results - Exp. transferability

- The agent in *Rooms* is pretty **slow** and **ineffective**
- The agent **is totally able** to walk down a *Corridor* ...
- ... but **not** if it's too *tight*
- In a *Turn* **collides once** every target
- In a *Crossroad* it finds its way but **collides too much**



Conclusions

For robotic locomotion and target localization tasks **reinforcement learning** and **curriculum learning** perform effectively.

The results show a different outcome based on the **evaluation method** chosen, underlining the difficulty to evaluate correctly reinforcement learning scenarios.

The models proposed seem to converge on a policy of **random search**, a behaviour shared between every experiment model. Most probably due to the **lack of memory** of the NN and **procedural generation** of the environment.

The models are able to generalize **obstacle avoidance**.

The models, seen the converged policy, manage to perform **target localization** discreetly.

Conclusions - Future works

- **Add memory to the system**: adding an RNN module (akin to the curiosity module) might allow the model to form an experience buffer of sorts and adopt smarter search policies.
- **Rework the reward and penalty functions**: implement more complex/effective functions, (eg soft-collisions)
- **Compare different RL algorithms**
- **Extend the types of scenarios**