A set of tools to build complex continuous control tasks for Deep Reinforcement Learning

Wilbert Pumacay Huallpa.

Universidad Católica San Pablo, Perú
wilbert.pumacay@ucsp.edu.pe

Abstract. Recent advances in Deep Reinforcement Learning were possible in part because of the simulators and benchmarks available, like the **Atari Learning Environment** (ALE) and **MuJoCo** based environments. These benchmarks offer great environments to train reinforcement learning agents, but there is a lack of diversity in the environments we can train on, as there are not enough tools to quickly build more complex environments on top of these (specially with the MuJoCo based environments for continuous control tasks). We propose a framework, which consists of a suite of tools to build complex environments on top of the **MuJoCo** physics engine (and **Bullet** for a free version), allowing researchers to create and standarize benchmarks for DeepRL agents in continuous control tasks.

Keywords: Deep Reinforcement Learning · Continuous control · Simulated Environments

1 Introduction

The field of Deep Reinforcement Learning (DRL) has achieved impressive results in tasks such as playing Atari games [11], beating the Go world champion [12], and winning against semi-proffesionals in the game of DOTA 2 [13]; and in recent years there have been various attempts to address continuous control tasks using DRL, which have yielded good results, such as [8], [9], and [10], just to name a few. The majority uses MuJoCo [7] as environment for DRL agent development.

As stated in [10], by defining rich and complex learning environments the trained agents can explore enough to learn complex skills. The contribution of this work consists of a framework to allow researchers to create these rich continuous control environments.

2 Related works

The main benchmarks used for continuous control tasks (as in [8], [9] and [10]) are built on top of the MuJoCo [7] physics engine, or some open source alternatives like Bullet or DART, as shown in the following table. Most of these environments currently support interaction with the physics engine and rendering support (if necessary), but lack the ability to **easily** create different types of complex environments, and most of the environments used as benchmarks are relatively simple (except of the ones used in [10], which mentioned the need of rich environments)

	Physics Engine	Needs license	Is wrapper	Reinforcement Learning API
MuJoCo's C API - [7]	MuJoCo	Yes	No	No
DeepMind Controlsuite - [1]	MuJoCo	Yes	Yes	Yes
OpenAI mujoco-py - [2]	MuJoCo	Yes	Yes	Yes
OpenAI Roboschool - [3]	Bullet	No	Yes	Yes
PyBullet-gym - [5], [6]	Bullet	No	Yes	Yes
DART-gym - [15], [16]	DART	No	Yes	Yes

Table 1: Comparison of benchmark alternatives for continuous control tasks

3 Proposed framework and Current progress

We propose a framework on top of current tools like [1] and [14] (as RL APIs), and [4] and [7] (as physics engines). The structure is shown in the following figure, and our contribution consists in the middle section shown in the figure below (**environment**, **agent** and **interaction** modules), a visualization tool (building on top of a custom 3d graphics engine) and the interfaces for the remaining components.

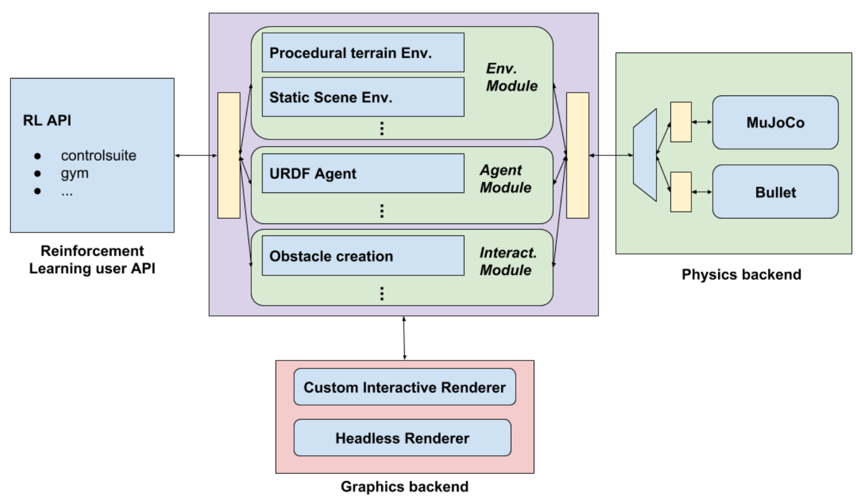

Fig. 1: Proposed framework

To assess the feasibility of the proposed framework we have started integrating on top of [1]. This focused on adding an interactive visualizer (graphics backend, as it was a missing feature), a nd tools for terrain generation. So far we have already integrated the visualizer (video found here).

The implementation can be found in our **fork**[1], which contains the integrated visualizer, a better documentation of the added API, and some examples of the usage..

[1] DeepMind's controlsuite fork: `https://github.com/wpumacay/dm_control`

References

1. Yuval Tassa, Yotam Doron, et al. DeepMind Controlsuite. ArXiv (2018)
2. OpenAI. mujoco-py. https://github.com/openai/mujoco-py
3. OpenAI. RoboSchool. https://github.com/openai/roboschool
4. Erwin Coumans. Bullet physics simulation. SIGGRAPH Courses (2015).
5. Pybullet. https://pybullet.org/wordpress/
6. PyBullet-gym. https://github.com/benelot/pybullet-gym
7. Emanuel Todorov, Tom Erez, Yuval Tassa. MuJoCo: A physics engine for model-based control. IROS (2012).
8. Timothy P. Lillicrap*, Jonathan J. Hunt*, et al. Continuous control with deep reinforcement learning. ICLR (2016).
9. John Schulman, Sergey Levine, et al. Trust Region Policy Optimization. ICML (2015)
10. Nicolas Heess, Dhruva TB, et al. Emergence of Locomotion Behaviours in Rich Environments. ArXiv (2017).
11. Volodymyr Mnih*, Koray Kavukcuoglu*, David Silver*, et al. Human-level control through deep reinforcement learning. Nature (2015).
12. David Silver*, Aja Huang*, et al. Mastering the game of Go without human knowledge. Nature (2017).
13. OpenAI Five. https://openai.com/five/
14. Greg Brockman, et al. OpenAI Gym. ArXiv (2016).
15. Lee et al. DART: Dynamic Animation and Robotics Toolkit. Journal of Open Source Software (2018).
16. OpenAI DART environments. https://github.com/DartEnv/dart-env/wiki/OpenAI-Gym-Environments.