Simulated Robotic Soft Body Manipulation

Extended Abstract

Glareh Mir
Institute for Artificial Intelligence
University of Bremen
Bremen, Germany
gmir@uni-bremen.de

Michael Beetz
Institute for Artificial Intelligence
University of Bremen
Bremen, Germany
beetz@uni-bremen.de

ABSTRACT

The performance of intelligent agents manipulating a soft body object depends on the agent's understanding of the execution environment. Hence, by keeping the agent fixed and changing the environment, the difference between environments can be measured. However, this becomes complicated when dealing with agents that learn in each environment.

We propose a framework for evaluating the influence of the simulated soft bodies (and related object models) on the reinforcement learning algorithms' performance. The change in algorithm behavior is quantified between different environments, and the correlation of behavioral difference is measured via statistical analysis. An evaluation case is presented on PyBullet and MuJoCo physics simulation environments with DDPG, PPO, TD3 and SAC algorithms.

KEYWORDS

Deep Reinforcement Learning; Learning and Adaptation; Soft Body Manipulation

ACM Reference Format:

Glareh Mir and Michael Beetz. 2024. Simulated Robotic Soft Body Manipulation: Extended Abstract. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 3 pages.

1 MOTIVATION

Prospection, the ability to anticipate future events, is an essential cognitive capability of intelligent agents. If an agent can foresee the result of their action, they can plan for handling failures, or even better, use failures to their advantage. However, increasing uncertainty in an environment could lead to variability in planning. In a closed non-deterministic environment, the level of uncertainty decreases as the experience of the manipulator (the intelligent agent) increases; leading to better planning and use of the environment. In other words, better planning is achieved when there is more certainty about the outcomes.

As an example of a closed non-deterministic environment, a domestic kitchen can be considered and explored. Disregarding external interference such as light condition, cats and kids, the scope of dealing with uncertainty is then reduced to dealing with manipulation of soft bodies in which the timing of interaction is of great importance. This is because a soft body can change shape in any step of interaction as opposed to a rigid body (imagine folding a silk handkerchief). Also, in reality soft bodies are very often slightly different in attributes such as size, shape, distribution of weight and density, etc. As such, no two pieces of ham are exactly the same in real world. Soft bodies that are closer to liquids in consistency are often managed through containers, therefore manipulated through rigid or soft body (squeezable) containers.

Therefore it is better to simplify the objective further by exploring a smaller scenario: a sandwich making scenario, where the robot must put a piece of ham (freely placed on its hand/gripper) on a piece of cheese and toast. The soft bodies are of different texture, and the ham is bendable. Here, simulation environments are used to explore methods of dealing with soft bodies and measure behavior, in a robust and environment-independent manner, enabling optimization of interaction without specifically providing motion parameters. This can be done through Deep Reinforcement Learning (RL).

Four methods of optimization via learning are used here; all belonging to the actor-critic RL methodologies: *Deep Deterministic Policy Gradient (DDPG)*[4], *Twin Delayed Deep Deterministic Policy Gradient (TD3)*[2], *Proximal Policy Optimization(PPO)*[6], and *Soft Actor-Critic (SAC)*[3]. This will provide us with the capability to abstract the simulated manipulation behavior and answer future-oriented questions such as "What trajectory parameters should I provide to successfully execute the manipulation task, knowing that the manipulated objects slightly differ from one another?", "How can we leverage minimal manipulation failures to our advantage in object manipulation?" or "How can it be called a job well done when the item does not stay in the target configuration for more than a few seconds?".

These questions can be interpreted more generally as "How many diverse prospection capabilities can be learned automatically,

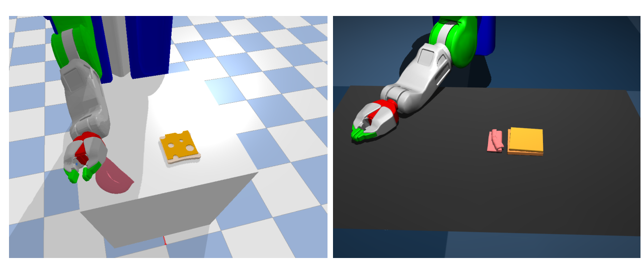

Figure 1: Physics Simulation Envs: PyBullet, MuJoCo

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

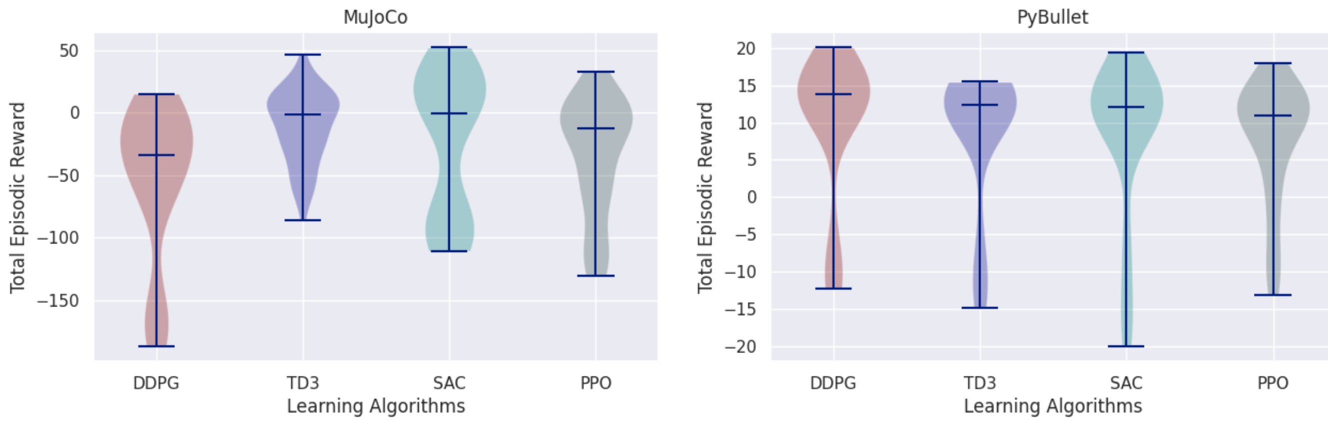

Figure 2: Reward distribution, presented as violin plots

and to what extent, from robot experiences in simulations?". However the scope of this question is currently beyond our capabilities. Instead let's focus on a sub-question: "How can the performance of models and learning algorithms in soft body manipulation be measured?". To answer this question, the design, requirements, and performance differences of the agents in the environments should also be explored.

It must be noted that dealing with soft bodies will result in execution that is very often not perfect; it is more about learning to control the object. An extreme example of this experience is a barehanded human holding a medium-size fish: the slipping, scales tearing off, and bending to the point of breaking that can happen if the fish is held too tight, too loose or from the wrong grasp point. The performance then cannot be measured by moving the fish from point A to point B without dropping it. A perfect grasp cannot be calculated quickly enough in this situation, so the human dynamically changes the goal to controlling the object. A simpler example could be children dealing with oily pasta.

A good starting point to answer the above question is to explore soft body representation in two heavily used physics engines: MuJoCo[7] and PyBullet [1]. They both have modules for representing soft bodies, particularly bendable bodies. The next step is to define which learning algorithms are used and on what models. It is important to reduce the variability in algorithms that the changes in behavior are *only* results of change in the learning environment and are independent from any type of hyperparameter. In doing so, the *total episodic reward* is measured and the *difference* between the results is compared; from each algorithm in each environment to the same difference in the other environment.

One particular advantage of using learning methods instead of hard-coding behavioral algorithms is the exploratory nature of learning. Allowing the agent to learn a behavior by itself results in child-like development of patterns of behavior that have not been observed before. The benefit of studying these emerging behaviors lies in the ability to compare the patterns that are unique to each learning and execution environment.

Given the exploratory nature of our methods, it would provide great analysis benefit if the input modalities for the RL action models are reduced to provide incentive for learning about the environment in a partially observable manner. Therefore, the input is reduced to only images from the robot's head mounted camera. It is suggested that this may have an impact on learning outcome [5], however it should restrict a great deal of learning about object properties from means other than visual image frames and provide blind spots that will significantly impact behavior and allow us to measure the difference of behavior by handycapping the action models through local maxima and minima.

2 CONCLUSION

We suggest a framework for comparison of performance of multiple reinforcement learning algorithms focusing on soft body object manipulation in two popular physics simulation environments.

The results achieved via this framework can point out issues to bear in mind for providing training facility for automatic training and learning of soft body manipulation, as well as how to measure consistency between environments for objectives such as transfer learning.

This framework focuses on different aspects of soft body manipulation required for intelligent prospection, as well as urging the scientists in the physics simulation fields to explore more realistic objects with better contact physics, to help us provide better prospection capabilities for robots.

ACKNOWLEDGMENTS

The research reported in this paper has been (partially) supported by the German Research Foundation DFG, as part of Collaborative Research Center (Sonderforschungsbereich) 1320 Project-ID 329551904 "EASE - Everyday Activity Science and Engineering", University of Bremen (http://www.ease-crc.org/). The research was conducted in subproject "R03: A knowledge representation and reasoning framework for robot prospection in everyday activity".

REFERENCES

[1] Erwin Coumans and Yunfei Bai. 2016–2021. PyBullet, a Python module for physics simulation for games, robotics and machine learning. http://pybullet.org.

[2] Scott Fujimoto, Herke Hoof, and David Meger. 2018. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*. PMLR, 1587–1596.

[3] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*. PMLR, 1861–1870.

[4] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[5] Glareh Mir, Matthias Kerzel, Erik Strahl, and Stefan Wermter. 2021. A humanoid robot learning audiovisual classification by active exploration. In *2021 IEEE International Conference on Development and Learning (ICDL)*. IEEE, Beijing, China, 1–6. https://doi.org/10.1109/ICDL49984.2021.9515598

[6] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[7] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 5026–5033. https://doi.org/10.1109/IROS.2012.6386109