Learning to Perform Physics Experiments via Deep Reinforcement Learning

Misha Denil[1] **Pulkit Agrawal**[2] **Tejas Kulkarni**[1]
Tom Erez[1] **Peter Battaglia**[1] **Nando de Freitas**[1,3,4]
{mdenil,tkulkarni,etom,peterbattaglia,nandodefreitas}@google.com
pulkitag@berkeley.edu
[1]DeepMind [2]University of California Berkeley [3]University of Oxford
[4]Canadian Institute for Advanced Research

Abstract

Object properties such as mass, friction, cohesion, and deformability often cannot be precisely determined through passive observation, yet people can easily infer these properties by interacting with them, just as a scientist discovers hidden facts about the world through experimentation. Recent advances in artificial intelligence have yielded machines that can achieve superhuman performance in Go, Atari, natural language processing, and complex control problems, but it is not clear that these systems can rival the scientific intuition of even a young child. Here we introduce a basic task that requires an agent to estimate objects' hidden properties, such as mass, in an interactive simulated environment in which they can manipulate the objects and observe the consequences. We found that cutting edge approaches in deep reinforcement learning can learn to perform the experiments necessary to discover the hidden properties. We also systematically manipulated the problem difficulty and the cost of experimenting, and found the agents learned different strategies that balanced the cost of gathering information against the gain of making correct estimates. These results open promising new directions toward creating machines that can not only explore and exploit, but explain.

1 Introduction

We would like to train agents that understand the world around them. Deep learning techniques in conjunction with vast labeled datasets have yielded powerful solutions to image and speech recognition. Deep reinforcement learning agents have also become very good at solving sequential decision tasks by learning deep video representations. However, we are interested in learning object properties such as mass, cohesion, and deformability that are typically acquired by interaction, and not by perception alone.

Computer vision researchers have realized that recognition performance can be improved by moving so as to acquire more views of an object or scene. There is a vast literature on active vision, see the related work section of Jayaraman and Grauman [2016]. Active exploration and physical interaction to learn visual representations has been effectively demonstrated in several recent works [Agrawal et al., 2016, Pinto et al., 2016]. Prediction from static images or video has also gained momentum [Mottaghi et al., 2016, Oh et al., 2015, Xue et al., 2016].

Humans can infer mass by watching movies of complex rigid body dynamics [Hamrick et al., 2016]. Using a physics engine, Wu et al. [2015] have shown that it is possible to learn properties such as mass from video. However, ample evidence [Smith and Gasser, 2005] provides us with strong motivation for studying and designing deep reinforcement learning agents capable of learning to interact with objects so as to answer questions about their properties.

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

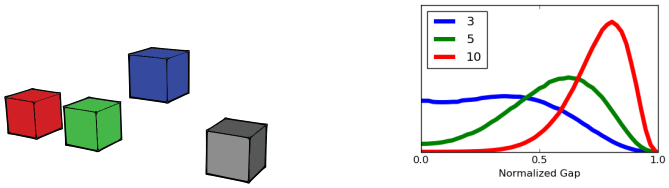

Figure 1: **Left:** Schematic diagram of the Which is Heavier environment. Agents can interact by applying upward force to one of the blocks. **Right:** Distributions of the mass gap for different settings of β.

2 Question answering through interaction

We present the Which is Heavier environment implemented on the MuJoCo physics engine for probing an agent's ability to learn about the non-visual property of object mass. A visualization of our environment is depicted in the left panel of Figure 1. The environment consists of four blocks, which are constrained to move only along the vertical axis. The four blocks are always the same size, but vary in mass between episodes.

The question to answer in this environment is which of the four blocks is the heaviest. Since the masses of the blocks are randomly assigned, the agent must poke the blocks and observe how they respond in order to find the heaviest one.

Controlling the mass distribution allows us to control the difficulty of the task. In particular, by controlling the size of the mass gap (i.e. the difference in mass between the two heaviest blocks) we can make the task more or less difficult. We make a distinction between *task* and *instance* level difficulty. The difficulty of an instance refers to the size of the mass gap for a particular assignment of masses to blocks, whereas the task difficulty refers to the tendency of a distribution over mass assignments to generate instances with a small mass gap.

We use the following generative process for assigning masses to blocks. First we select one of the blocks uniformly to be the "heavy" block and designate the remaining three as "light." We sample the mass of the heavy block from $\mathrm{Beta}(\beta, 1)$ and the mass of the light blocks from $\mathrm{Beta}(1, \beta)$. Varying the parameter β allows us to control the distribution of mass gaps in a flexible way. We show the mass gap distribution, normalized to the range of possible masses, for different values of β in the right panel of Figure 1.

Agents progress through the environment in three stages.

Interaction Initially there is an exploration phase, where the agent is free to interact with the environment and gather information.

Labeling At the end of the exploration phase the agent produces a *labeling* action through which it indicates which block it has chosen as the heaviest.

Reward The environment responds to a labeling action with a reward, +1 for the correct answer and -1 for incorrect, and the episode terminates.

Crucially, the transition between interaction and labeling does not happen at a fixed time step, but is actually initiated by the agent. The full set of interaction and labeling actions are available to the agent at every time step, and it initiates the transition to the labeling phase simply by selecting a labeling action. This allows the agent to decide when enough information has been gathered, but also forces the agent to balance the trade-off between answering now given its current knowledge, or delaying its answer to gather more information.

The optimal trade-off between information gathering and risk of answering incorrectly depends on two factors. The first is the difficulty, which we control by varying β, and the second is the *cost of information*, which can be controlled by varying the discount factor during learning. A small discount places less emphasis on future rewards, and encourages the agent to produce a label quickly.

Because the reward stage is only reached when the agent takes a labeling action, there is a degenerate case where the agent learns to never produce a label and thereby never receive a negative return. To avoid this solution we impose a time limit on episodes and terminate with a reward of -2 if this limit is reached.

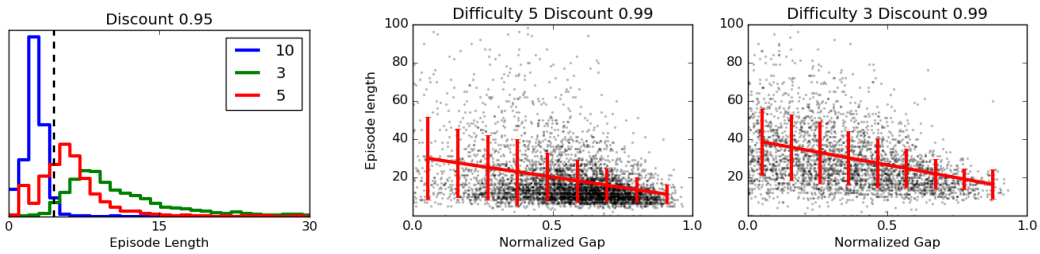

Figure 2: **Left:** Histograms of episode lengths for different task difficulty (β) settings. There is a transition from $\beta = 10$ where the agents answer eagerly as soon as they find a heavy block to $\beta = 3$ where the agents are more conservative about answering before they have acted enough to poke all the blocks at least once. **Right:** Episode lengths as a function of the normalized mass gap. Units on the x-axes are scaled to the range of possible masses, and the y-axis shows the number of steps before the agent takes a labeling action. The black dots show individual episodes, and the red line shows a linear trend fit by OLS and error bars show a histogram estimate of standard deviations. Each plot shows the testing episodes of a single trained agent.

In this paper we consider the simplest possible action space for this environment, which is to provide actions that allow the agent to affect forces directly on each object. We give the agent a discrete action for each block, such that taking an action causes an upwards force to be applied to the corresponding block. In addition to the four interaction actions the agent also has four labeling actions that indicate a guess at the heaviest block.

3 Experiments

We present two experiments showing how problem difficulty leads to differentiated behavior both at the agent level and at the problem instance level. The first experiment shows that as we increase the problem difficulty the learned policies transition from guessing immediately when a heavy block is found to strongly preferring to poke all blocks before making a decision.

The second experiment shows that we also see behavior adapting to problem difficulty on the level of individual problem instances, where a single agent will tend to spend longer gathering information when the particular problem instance is more difficult.

Population strategy differentiation For this experiment we trained agents at three different difficulties corresponding to $\beta \in \{3, 5, 10\}$ all using a discount factor of $\gamma = 0.95$ which corresponds a relatively high cost of gathering information. We trained three agents for each difficulty and show results aggregated across the different replicas.

After training each agent was run for 10,000 steps under the same conditions they were exposed to during training. We record the number and length of episodes executed during the testing period as well as the outcome of each episode. Episodes are terminated by timeout after 100 steps, but the vast majority of episodes are terminated in < 30 steps by the agent producing a label. Since episodes vary in length not all agents complete the same number of episodes during testing. Overall success rate on this task is high, with the agents choosing correctly on 96.3% of episodes.

The left plot in Figure 2 shows histograms of the episode lengths broken down by task difficulty. The red vertical line indicates an episode length of four interaction steps, which is the minimum number of actions required for the agents to interact with every block. At a task difficulty of $\beta = 10$ the agents appear to learn simply to search for a single heavy (which can be found with an average of two interactions). However, at a task difficulty of $\beta = 3$ we see a strong bias away from terminating the episode before taking at least four exploratory actions.

Individual strategy differentiation For this experiment we trained agents using the same three task difficulties as in the previous experiment, but with an increased discount factor of $\gamma = 0.99$. This decreases the cost of exploration and encourages the agents to gather more information before producing a label, leading to longer episodes.

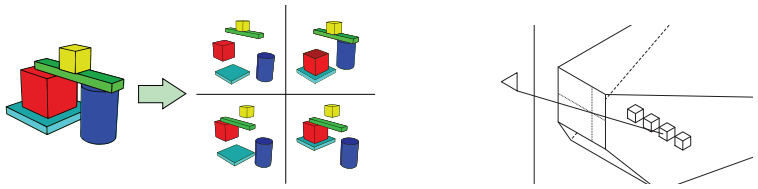

Figure 3: **Left:** The underlying structure of the proposed Towers environment. **Right:** Schematic diagram of the Point and Throw action space. The agent controls a virtual "cursor" that can be used to poke or select the underlying object in the scene.

After training each agent was run for 100,000 steps under the same conditions they were exposed to during training. We record the length of each episode, as well as the mass gap between the two heaviest blocks in each episode. In the same way that we use the distribution of mass gaps as a measure of task difficulty, we can use the mass gap in a single episode as a measure of the difficulty of that specific problem instance. We again exclude from analysis the very small proportion of episodes that terminate that timeout.

The right plots in Figure 2 show the relationship between the mass gap and episode length across the testing runs of two different agents. From these plots we can see how a single agent has learned to adapt its behavior based on the difficulty of a single problem instance. Although the variance is high, there is a clear correlation between the mass gap and the length of the episodes.

4 Ongoing work

We are working on extending the scope of our experiments to include additional environments, actuators and perceptual modalities. New environments will allow us to ask questions about different non-visual properties of objects, and different types of actuators will allow us the agents to engage with objects in their environment in more interesting ways. Finally, learning from pixels rather than features will allow us to probe visual understanding more directly.

On the environment front we are building an environment in which the agent is presented with a tower composed of small blocks, which could decompose into rigid objects in a number of different ways that cannot be distinguished without poking it. A schematic of this environment is shown on the left of Figure 3. The task in this environment is to count the number of rigid bodies, which will allow us to study how agents can form an understanding of object cohesion.

In order to provide more interesting actuators, while avoiding the problem of dexterous manipulation, we are developing the Point and Throw actuator, where the agent controls the position of a virtual "cursor" that can be used to "click" on objects in order to apply forces or to indicate them as the target of a labeling action. This is depicted in the right of Figure 3.

References

P. Agrawal, A. Nair, P. Abbeel, and J. Malik. Learning to poke by poking: Experiential learning of intuitive physics. In *Neural Information Processing Systems*, 2016.

J. B. Hamrick, P. W. Battaglia, T. L. Griffiths, and J. B. Tenenbaum. Inferring mass in complex scenes by mental simulation. *Cognition*, 157:61–76, 2016.

D. Jayaraman and K. Grauman. Look-ahead before you leap: End-to-end active recognition by forecasting the effect of motion. In *European Conference on Computer Vision*, pages 489–505, 2016.

R. Mottaghi, M. Rastegari, A. Gupta, and A. Farhadi. "what happens if..." learning to predict the effect of forces in images. In *European Conference on Computer Vision*, pages 269–285, 2016.

J. Oh, X. Guo, H. Lee, R. Lewis, and S. Singh. Action-conditional video prediction using deep networks in Atari games. In *Neural Information Processing Systems*, pages 2863–2871, 2015.

L. Pinto, D. Gandhi, Y. Han, Y. Park, and A. Gupta. The curious robot: Learning visual representations via physical interactions. In *European Conference on Computer Vision*, pages 3–18, 2016.

L. Smith and M. Gasser. The development of embodied cognition: Six lessons from babies. *Artificial life*, 11 (1-2):13–29, 2005.

J. Wu, I. Yildirim, J. J. Lim, B. Freeman, and J. Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In *Neural Information Processing Systems*. 2015.

T. Xue, J. Wu, K. L. Bouman, and W. T. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. *arXiv preprint arXiv 1607.02586*, 2016.