Using an Energy-Based Model to provide rewards to a Reinforcement Learning agent for the control of a robot arm.

Peter Coates

April 2023

PROBLEM DESCRIPTION

Given a simple description of object properties and relationships as a goal, e.g. ‘blue cube on red disc’, can a robot be taught using reinforcement learning to position the objects as requested?

An agent using reinforcement learning discovers which actions yield the most reward by trying them (Sutton and Barto, 2020). So, for a robot to be taught how to reach a goal like ‘blue cube on red disc’ via reinforcement learning, it needs the freedom to try many actions, and a reward function that can lead it to the goal.

Zhu et al (2020) use a version of the soft actor critic algorithm (described by Haarnoja et al, 2018) to demonstrate that it is possible to train a robot arm to perform positioning tasks with the only human intervention being the provision of images showing the goal state. An approach called variational inverse control with events (VICE) proposed by Fu et al (2018) can generate rewards from just the end goal. It generalises inverse reinforcement learning and considers the case where only the desired final outcome is known. Zhu et al (2020) successfully used the VICE framework to learn a discriminator from the goal images which then generates rewards for the reinforcement learning agent.

A generative adversarial network (Goodfellow et al, 2014) trained on suitable images would be able to generate sample images from a goal description such as ‘blue cube on red disc’. These could then be used as the goal images for training the robot. Yu, Nan and Ku (2021) used a generative adversarial network to generate multiple training samples with which they successfully trained for robot hand-eye cooperation using inverse reinforcement learning.

However, Ramesh et al (2021) show that approaches like generative adversarial networks are not good at encoding the relationships between objects. An alternative approach used by Liu et al (2021) is to encode relationships using Energy-Based Models which are then used to reliably generate images from those relationships.

This project will use an Energy-Based Model (LeCun et al, 2006) to generate a reward function that can use descriptions such as ‘blue cube on red disc’ to provide rewards for actions taken by a reinforcement learning agent controlling a robot arm simulation to position objects as requested in the original description.

PROJECT OBJECTIVES

The main objective is to create a system that positions objects using a simple description of the objects and their relationship to each other. Combining previous work by Liu et al (2021) and Zhu et al (2020) suggests that this may be possible.

The project will use a simulated environment for both training and testing. There are a number of advantages gained from using a simulated environment rather than the real world:

- Easily reset to known starting points.
- More actions can be performed in the same time frame.
- No risk of damage / failure to equipment.

Lobbezoo, Qian and Kwon (2021) point out that the majority of recent research uses simulated environments, and that experiments using real world systems have proved to be less accurate than experiments using simulations. It is important to remember that to be useful, the robot controller would need to operate in the real world. This project is not attempting to answer the question of how results in a simulated environment can successfully be transferred to the real world. However, it is looking to create a reward function that could easily be applied in either a simulated or real environment.

The main objective of this project is challenging and relies on the successful completion of a number of components, namely:

- Simulated Environment
- Energy-Based Model
- Reward Function derived from the Energy-Based Model
- Reinforcement Learning agent

Once these components have been created they need to be integrated together and their ability to perform positioning tasks needs to be measured.

The approach to integrate the components will be to create a ‘hand eye coordination game’ using the simulated environment and the reward function. The game will have an API in line with the Gymnasium API (Farama Foundation, 2022) which will accept an action, apply it to the simulated environment and return an observation of the result (an image of the environment) and a reward from the reward function. This will allow it to be controlled by both a human player, and a reinforcement learning agent.

The requirements for the hand eye coordination game include:

- Have a number of goal tasks
- Select a task to be played
- Scores based on how long it takes to complete the selected task and rewards received
- Limit the available time to complete tasks
- Once task complete or run out of time, reset environment and give another task
- Tasks can be given different viewing positions
- Environment can be set up with different numbers of objects in it

The game will initially by played by a human player. This will allow the reward function to be empirically tested and a baseline to be established for a reasonable score. A reinforcement learning agent can then play the game and be assessed against the human player.

This approach allows various parts of the system to be altered independently, and the success of those alterations to be measured.

For example, different algorithms can be used to create the reinforcement learning agent. Zhu et al (2020) successfully used the soft actor critic algorithm, and this will be used as the starting point for this project. This will then be compared against other algorithms. Exactly which algorithms will be determined during the project, however, deep deterministic policy gradient (DDPG) and deep Q-Networks (DQN) have been used in a number of other research projects (Franceschetti et al, 2022. Lobbezoo, Qian and Kwon, 2021. Elguea-Aguinaco et al, 2023) and appear to be good candidates.

The reward function is another area that different research papers approach in different ways. Lobbezoo, Qian and Kwon (2021) include a section in their survey explaining a number of techniques used for reward shaping. These are techniques that breakdown the main task so that rewards can be applied more frequently. Often resulting in faster learning, but at the expense of accuracy. Although not explicitly mentioned in the plan below, investigating different reward shaping techniques will be included in the project if time permits.

The following table lists the components to be created by this project with a brief overview of how each will be achieved:

Component	**Overview**
Simulated Environment	The simulated environment needs to contain both a robot arm and a number of objects and requires an interface that makes the following available: 1. Views of the objects in different arrangements (e.g. one on top of another, or side by side) 2. The ability for the robot arm to move objects within the simulated environment. Coppelia Robotics (2023) provide virtual simulations that interface with the Mujoco physics engine (Todorov et al, 2012). Using their software should help generate a suitable simulated environment.
Energy-Based Model	Liu et al (2021) used a number of different data-sets to train an Energy-Based Model. These data-sets are publicly available (see resources section for details) and may be used along with the simulated environment for this project. The Energy-Based Model will be trained to capture the relationships of objects in the simulated environment so that it can both identify individual objects, and a group of objects that are related to each other. For example, ‘blue cube on red disc’.
Reward Function	Use the VICE framework (Fu et al, 2018) or similar to create the reward function. The trained Energy-Based Model will be used to generate suitable rewards. This may take the form of generating sample goal images.
Hand-eye coordination game	The simulated environment and reward function will be integrated to form a game which takes an action as an input and produces an observation (an image of the environment) and a reward as output. It will initially be played by a human to assess how well the reward function works, and determine a baseline score that can be used to measure the efforts of different reinforcement learning agents.
Reinforcement Learning agent	This will play the hand eye coordination game. Initially the soft actor critic algorithm (Zhu et al, 2020) will be used.
Alternative RL agents	The performance of the soft actor critic algorithm will be compared against other algorithms. The exact algorithms will be determined during the project. DDPG and DQN appear to be good candidates.

At first site the components being generated may appear to be re-implementations of existing work. Although this has its own value in that it will validate recent research, there are also new lines of investigation here. The author has not yet found any research that generates reward functions directly from Energy-Based Models.

Also, the experiments used by Zhu et al (2020) restricted the movement of objects to a single dimension. There is no guarantee that their approach will be applicable in a three dimensional space.

PROJECT PLAN

SCRUM (Schwaber and Sutherland, 2020) is a good project methodology, but is focused on a project team rather than working as an individual. The main principle of SCRUM is to work in short blocks of time called sprints. A sprint would typically be between two and four weeks. Each sprint starts by planning what tasks are to be performed during the sprint. There’s a team catch-up every day to talk through what is going to get done that day, and how things are going. At the end of the sprint the outputs from all the tasks are delivered. Finally there’s a retrospective to review how the sprint went and suggest improvements for the next one.

These basic principles of SCRUM will be applied to this project.

11 four week sprints will be used. And each week will have 13 to 14 study hours, giving a total of approximately 600 study hours for the project. Using 11 four week sprints for the 12 month project gives flexibility to include breaks as and when required.

The outputs from each sprint will be published to a GitHub repository (Coates, 2023). This will provide both a safe copy of work done, and a good indication of project progress. Part of the output for each sprint will be updates to both the literature review and the dissertation report. The literature review will see most activity early in the project, but will be an ongoing task during the project.

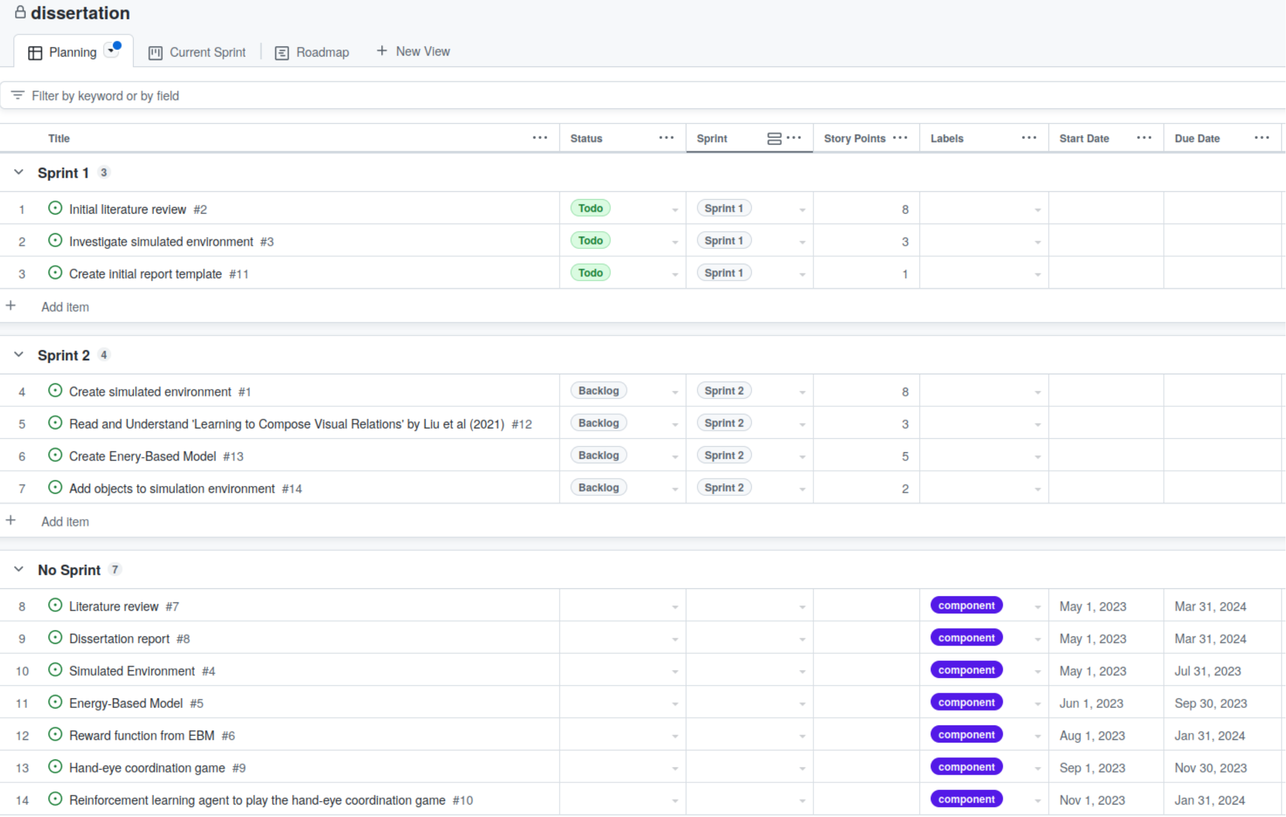
The following roadmap shows how work on the various components will be spread across the lifetime of the project. To keep things simple, the sprints have been mapped to months.

<table>
<tr><th>Year</th><th colspan="7">2023</th><th colspan="4">2024</th></tr>
<tr><th>Month</th><th>May</th><th>Jun</th><th>Jul</th><th>Aug</th><th>Sep</th><th>Oct</th><th>Nov</th><th>Dec</th><th>Jan</th><th>Feb</th><th>Mar</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><th>Sprint</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th></tr>
<tr><td></td><td colspan="11">Literature Review</td></tr>
<tr><td></td><td colspan="11">Dissertation Report</td></tr>
<tr><td></td><td colspan="3">Simulated Environment</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="4">Energy-Based Model</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td colspan="6">Reward Function</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td colspan="3">Hand-eye coordination game</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="3">RL Agent</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="3">Alternative Agents</td></tr>
</table>

Figure 1 : Roadmap showing when components are to be worked on during the project.

Issues have been created in the GitHub repository for each of the components required. These issues are then broken down into lists of tasks, with each task being assigned a number of story points to indicate its size. The tasks will then be added to a sprint so that they can be listed and tracked in the project on GitHub.

The tasks for the first two sprints have already been added to the project, and the following is a screenshot from the GitHub project:



dissertation

Planning | Current Sprint | Roadmap | + New View

Filter by keyword or by field

	Title	Status	Sprint	Story Points	Labels	Start Date	Due Date
	Sprint 1 3						
1	Initial literature review #2	Todo	Sprint 1	8			
2	Investigate simulated environment #3	Todo	Sprint 1	3			
3	Create initial report template #11	Todo	Sprint 1	1			
+	Add item						
	Sprint 2 4						
4	Create simulated environment #1	Backlog	Sprint 2	8			
5	Read and Understand 'Learning to Compose Visual Relations' by Liu et al (2021) #12	Backlog	Sprint 2	3			
6	Create Enery-Based Model #13	Backlog	Sprint 2	5			
7	Add objects to simulation environment #14	Backlog	Sprint 2	2			
+	Add item						
	No Sprint 7						
8	Literature review #7				component	May 1, 2023	Mar 31, 2024
9	Dissertation report #8				component	May 1, 2023	Mar 31, 2024
10	Simulated Environment #4				component	May 1, 2023	Jul 31, 2023
11	Energy-Based Model #5				component	Jun 1, 2023	Sep 30, 2023
12	Reward function from EBM #6				component	Aug 1, 2023	Jan 31, 2024
13	Hand-eye coordination game #9				component	Sep 1, 2023	Nov 30, 2023
14	Reinforcement learning agent to play the hand-eye coordination game #10				component	Nov 1, 2023	Jan 31, 2024

Figure 2 : Initial Sprint breakdown in GitHub project.

The tasks for future sprints will be added to the project during the sprint planning which takes place at the start of the sprint. At the same time, progress will be reviewed against the roadmap shown in Figure 1. If progress is ahead of the roadmap, then extra tasks can be brought into a sprint. If progress is behind, then it provides an early indication that the scope of the project may need to be reassessed.

RESOURCES

Liu et al (2021) used labeled data-sets of multiple objects and scene description to train Energy-Based Models. These data-sets are publicly available and may be used to train the Energy-Based Model.

- CLEVR https://cs.stanford.edu/people/jcjohns/clevr/ (Johnson et al, 2016)
- iGibson https://svl.stanford.edu/igibson/ (Shen et al, 2021)

Further training data will be created as part of the project from the simulated environment. This will likely use CoppeliaSim (Coppelia Robotics, 2023) which has the ability to generate simple objects and view them from different locations using a virtual camera. Figure 3 shows a simple example of images from CoppeliaSim showing ‘a blue cube on a red disk’ from two very different viewing angles.

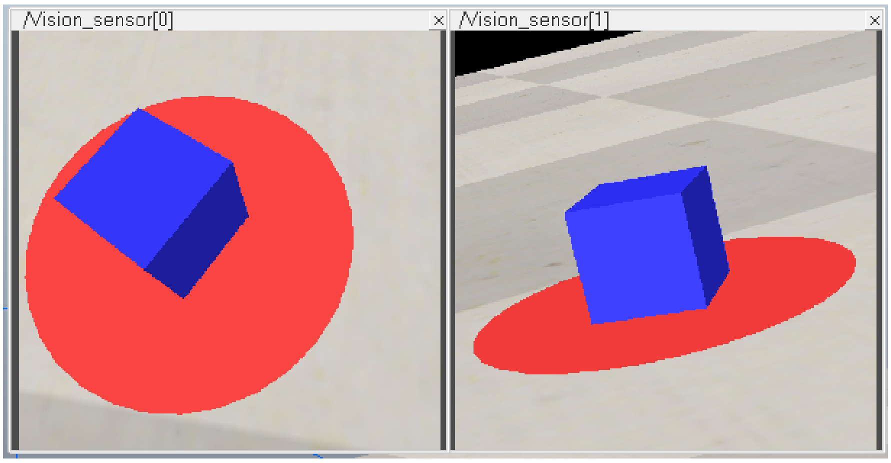


Figure 3 : Blue cube on a red disc in CoppeliaSim.

A free educational license is available for CoppeliaSim.

The MuJoCo framework is licensed under the Apache License, Version 2.0 (Apache Software Foundation, 2004).

Hardware : a dedicated desktop PC will be used to run the simulations. If extra processing is required, options to use cloud resources will be investigated.

REFERENCES

Apache Software Foundation, 2004. Apache License, Version 2.0 [Online]
Available from: https://www.apache.org/licenses/LICENSE-2.0 [Accessed 26 Mar 2023]

Coates, P., 2023. AI Msc Dissertation GitHub repository [Online]
Available from: https://github.com/pcoates33/ai-msc-dissertation [Accessed 21 April 2023]

Coppelia Robotics, 2023. Dynamics Module [Online]
Available from: https://www.coppeliarobotics.com/helpFiles/en/dynamicsModule.htm [Accessed 26 Mar 2023]

Elguea-Aguinaco, Í., Serrano-Muñoz, A., Chrysostomou, D., Inziarte-Hidalgo, I., Bøgh, S. and Arana-Arexolaleiba, N., 2023. A review on reinforcement learning for contact-rich robotic manipulation tasks. Robotics and Computer-Integrated Manufacturing [Online], 81, p.102517.
Available from: https://doi.org/10.1016/j.rcim.2022.102517 [Accessed 21 Apr 2023]

Farama Foundation, 2022. Gymnasium Documentation [Online]
Available from: https://gymnasium.farama.org/ [Accessed 21 April 2023]

Franceschetti, A., Tosello, E., Castaman, N., Ghidoni, S., 2022. Robotic Arm Control and Task Training Through Deep Reinforcement Learning. Springer [Online].
Available from: https://link.springer.com/chapter/10.1007/978-3-030-95892-3_41 [Accessed 26 Mar 2023]

Fu, J., Singh, A., Ghosh, D., Yang, L., Levine, S., 2018. Variational inverse control with events: A general framework for data-driven reward definition. ArXiv [Online], 1805.11686
Available from: https://doi.org/10.48550/arXiv.1805.11686 [Accessed 26 Mar 2023]

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative Adversarial Nets (PDF). Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014).
Available from: https://arxiv.org/abs/1406.2661 [Accessed 26 Mar 2023]

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, A., Sergey Levine, S., 2018. Soft actor-critic algorithms and applications. ArXiv [Online], 1812.05905.
Available from: https://arxiv.org/abs/1812.05905 [Accessed 26 Mar 2023]

Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C.L. and Girshick, R., 2016. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning [Online].
Available from: https://doi.org/10.48550/ARXIV.1612.06890 [Accessed 21 Apr 2023]

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, A. and Huang, F.J., 2006. A Tutorial on Energy-Based Learning. Predicting Structured Data, MIT Press [Online].
Available from: http://yann.lecun.com/exdb/publis/pdf/lecun-06.pdf [Accessed 21 Apr 2023]

Liu, N., Li, S., Du, Y., Tenenbaum, J.B., Torralba, A., 2021. Learning to Compose Visual Relations. ArXiv [Online], 2111.09297.
Available from: https://arxiv.org/abs/2111.09297 [Accessed 26 Mar 2023]

Lobbezoo, A., Qian, Y. and Kwon, H.-J., 2021. Reinforcement Learning for Pick and Place Operations in Robotics: A Survey. Robotics [Online], 10(3), p.105.
Available from: https://doi.org/10.3390/robotics10030105 [Accessed 21 Apr 2023]

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I., 2021. Zero-shot text-to-image generation. arXiv preprint arXiv:2102.12092, 2021

Schwaber, K., Sutherland, J., 2020. The Scrum Guide. scrumguides.org [Online].
Available from: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf [Accessed 26 Mar 2023]

Shen, B., Xia, F., Li, C., Martin-Martin, R., Fan, L., Wang, G., Buch, C.P.-D.S., Srivastava, S., Tchapmi, L., Tchapmi, M., Vainio, K., Wong, J., Fei-Fei, L. and Savarese, S., 2021. iGibson 1.0: A Simulation Environment for Interactive Tasks in Large Realistic Scenes. 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) [Online], September 2021. IEEE.
Available from: https://doi.org/10.1109/iros51168.2021.9636667 [Accessed 21 Apr 2023]

Sutton, R.S., Barto, A.G., 2018. Reinforcement Learning An Introduction. 2nd ed. London, The MIT Press.

Todorov, E., Erez, T., Tassa, Y., 2012. MuJoCo: A physics engine for model-based control. IEEE [Online].
Available from: https://github.com/deepmind/mujoco [Accessed 26 Mar 2023]

Zhu, H., Yu, J., Gupta, A., Shah. D., Kristian, H., Singh, A., Vikash, K., Levine, S., 2020. The Ingredients of Real-World Robotic Reinforcement Learning. ArXiv [Online], 2004.12570
Available from: https://doi.org/10.48550/arXiv.2004.12570 [Accessed 26 Mar 2023]

Yu, N., Nan, L. and Ku, T., 2021. Robot hand-eye cooperation based on improved inverse reinforcement learning. Industrial Robot: the international journal of robotics research and application [Online], 49(5), pp.877–884.
Available from: https://doi.org/10.1108/ir-09-2021-0208 [Accessed 4 Apr 2023].