Causal Reasoning for Robot Manipulation using the CausalWorld Framework

Kausik Lakkaraju
Department of Computer Science and Engineering
University of South Carolina

I. Abstract

Causal Reasoning is one of the most crucial cognitive skills that humans possess and has been a major part of evolution. This reasoning will also be a very crucial aspect in the development of intelligent robots [6]. Many environments like Gym [14] that were created in the past for training Reinforcement Learning (RL) agents. However, there is no notion of causation in these environments and it is not known to what degree the agents can be expected to generalize. Being able to intervene on different variables in the environment while training helps the agents to generalize well. This would make the system perform tasks which are sampled from out-of-distribution-data (OOD) as well. CausalWorld [1] framework which was proposed in 2020 is one of the frameworks that allows us to perform various interventions on the variables environment. My aim is to examine the capabilities and limitations of the framework by making use of the provided functionality and present my results. Some demonstrations that I recorded can be found here and my code for implementation can be found here.

II. Introduction

"... you are smarter than your data. Data do not understand causes and effects; humans do."

- Judea Pearl, *The Book of Why*

One of the most important elements that AI lacks is commonsense reasoning or reasoning, in general. In the recent years, there has been a growth in research in this area. Causal Reasoning is at the intersection of Philosophy and Economics and it identifies the relationship between a cause and its effect. However, its applications are not limited to these domains but can be extended to several other domains like computer science and healthcare.

According to Paul Mackenzie and Judea Pearl [11], there are three rungs in what they called, 'the ladder of causation'.

- **Observational**: Many of the existing systems in AI and Robotics domains which make use of data to train their systems use observational data. This is the bottom most rung of the ladder of causation.
- **Interventional**: This makes use of the notion of intervention where the user or the system intervenes on the variables present in the environment or the data. Intervention is the process of forcing a certain variable to take a specific value rather than just observing the variable. There are various means through which we can perform this intervention. In the robotics domain, if we are given an environment, we could intervene on certain parameters in the environment like changing the shape of objects the robot needs to move, goal positions, or changing physical attributes of certain objects in the environment.
- **Counterfactual**: Counterfactual reasoning tries to answer questions like ”What if ...?” and ”If they had only ...”. How things would have turned out if one had chosen to do something else rather than what they actually did?

With the framework, [1], it is possible to simulate a robot which could do multiple tasks in a given environment. The framework allows us to make use of the given two arms of the robot to perform several tasks like stacking, pushing, pick and place, etc. The most interesting contribution of the framework is the ability to do interventions on various attributes of the environment including the ability to generate different goal positions by intervening.

III. Framework Description

- CausalWorld framework provides us with eight different tasks that the robot could perform in the given environment. Snapshots of the environment can be seen in Fig. 1 and Fig. 2.
- Different trained policies are provided to solve some of the tasks. All of the policies except 'grasping policy' are trained on some observations. Grasping policy was handmade i.e. there was no learning involved.
- The framework allows us to intervene on different variables in the environment by changing their orientation, position or, in some cases, even their physical attributes like size, shape and color. This would augment the data that is needed for the model to get trained efficiently.
- The framework allows you to create a task from scratch, train a policy from scratch and use this trained policy to solve the created task. This shows how flexible the framework is.

IV. Problem Description

The aim of this project is to:

- Evaluate some of the given tasks by training the policies required to solve them from scratch.
- Evaluate some of the existing policies across the given protocols.

- To examine the flexibility and efficiency of the framework by creating a task from scratch and training a policy to solve it. Finally, I will try to solve the created task using this trained policy.
- To examine the capabilities and limitations of the framework by making effective use of the provided functionality.

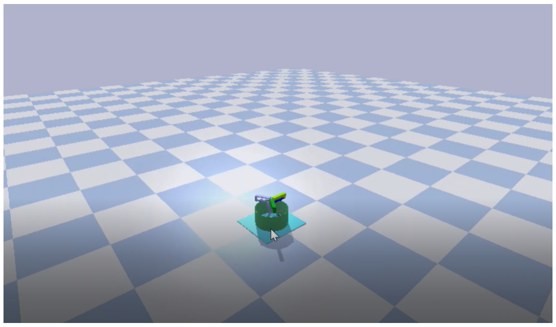


Fig. 1. The CausalWorld environment

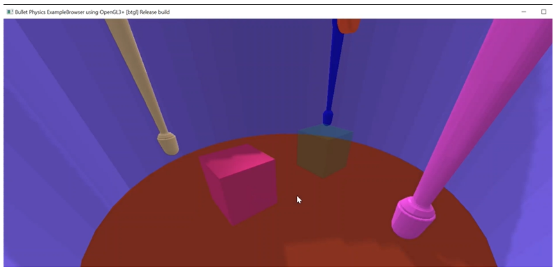
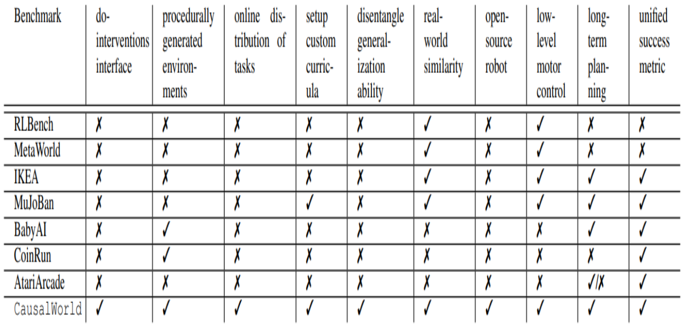

Bullet Physics ExampleBrowser using OpenGL3+ [btgl] Release build

Fig. 2. A closer look at the CausalWorld environment

V. Related Work

In [1], the authors proposed a benchmark for causal structure and transfer learning called *CausalWorld* which is the basis for this project. The tasks consists of constructing 3D structures from a given set of blocks or to move the end-affectors from one point to another and the framework also allows users to intervene on many variables (the Do-calculus) in the environment and observe the changes in other variables. The tool also allows us to create our own custom tasks using underlying functionalities. The robot in the world is similar to [2], which is a tri-finger robot. The ability to intervene not only helps us to identify the causal relation between different variables in the environment but also gives users the ability to collaborate with the robot in solving certain tasks.

In [1], the authors have also given a very useful table comparing features of different environments with CausalWorld. You can see this in Fig. 3.



Benchmark	do-interventions interface	procedurally generated environments	online distribution of tasks	setup custom curricula	disentangle generalization ability	real-world similarity	open-source robot	low-level motor control	long-term planning	unified success metric
RLBench	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗
MetaWorld	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗
IKEA	✗	✗	✗	✗	✗	✓	✗	✓	✓	✓
MuJoBan	✗	✗	✗	✓	✗	✓	✗	✓	✓	✓
BabyAI	✗	✓	✗	✗	✗	✗	✗	✗	✓	✓
CoinRun	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓
AtariArcade	✗	✗	✗	✗	✗	✗	✗	✗	✓/✗	✓
CausalWorld	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Fig. 3. Comparison of CausalWorld with other RL Benchmark environments

One might ask: Why do we need to introduce causality? Why couldn't one make use of the existing RL models or data-centric machine learning models to solve problems in Robotics and AI? Most of the existing models and models that are being built assume that two variables are correlated but does not take into account the actual causal relation between these. Udny Yule, a British statistician, and Karl Pearson, English mathematician, introduced concepts of correlation and regression in 1900s. However, Yule and Pearson found some situations where they found this correlation inference not very satisfactory [4]. This is because ”correlation does not imply causation”. This is a trending mantra in the research community which has paved way for more robust and transparent systems. Moreover, training the robot in an adversarial fashion makes it more robust. Human interventions can be mimicked in such a way as to create an attack and observe how the robot reacts to it.

To make this point even more stronger, let us consider an example that was mentioned in [6]. There is a large part of robotics research addressing how a robot interprets human actions or tries to read a human's mind observing their actions. For example, if a robot at an elder care facility is taking out a juice bottle from a fridge, the robot recommends the user to take blood sugar test. The robot assumed that the elderly person is thirsty. This is called spurious correlation. The person must have taken the bottle to give it another person or to discard it as it is past its expiry date or he might be just cleaning the fridge and arranging things. This kind of causal reasoning is required in robots right now. There was a previous work which did this [7]. The paper tries to achieve two problems: intent recognition and adaptation i.e. the robot should be able to determine the causal link to understand the intention of user behind a particular action and simultaneously assessing the possible actions that robot could take consistently.

In [8], authors used causal relations for robot imitation and plan recognition. From their experiments, they have concluded that causal reasoning is effective and useful for imitation learning. Another interesting work in this area is [5] where the authors tried to generate counterfactuals like ”Can we change event A such that it caused B instead of C?” thereby trying to make the system more transparent.

The ”do-analysis” which comes with human intervention and ”what if...?” analysis which comes with counterfactual reasoning allows us to observe the relationship between variables in an environment, avoid spurious correlations and to see the cases where robot fails and cases in which robot succeeds in performing the given task. Such experiments would help us improve the causal models thereby making the systems more robust and transparent.

VI. Approach

I followed a specific workflow to solve the problem that I defined. Fig. 4 shows the workflow diagram.

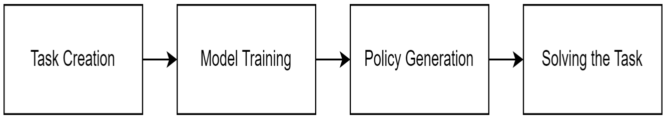


Fig. 4. Workflow diagram

1) *Task Creation*: Some of the tasks that are provided in the framework are towers, stacking, pushing, picking, reaching, etc. We can also create our own task by using their 'base_task' function. I have created two tasks by modifying their existing ones.
2) *Model Training*: I trained the models using PPO2 algorithms provided by 'stable_baseline' library. PPO (Proximal Policy Optimization) [12] is a class of Reinforcement Learning algorithms which are easy to implement and tune. They compute an update at each step that minimizes the cost function while making sure that it does not deviate much from the previous policy. OpenAI released a variant of PPO called PPO2 which is a GPU-enabled implementation. It runs 3 times faster than the vanilla PPO on Atari [13]. There are various policies that can be used with PPO2. I used 'MlpPolicy' object. It implements actor critic using an MLP (Multi-Layer Perceptron) with 2 layers of 64 neurons in total.
3) *Policy Generation*: CausalWorld provides us with various trained policies like grasping policy, picking policy, etc. to solve some of the existing tasks. It also provides a 'base_policy' which is a wrapper that can be used to create policies for our own trained models.
4) *Solving the task*: The final step in the workflow is to solve the created task using the trained policy.

VII. Implementation

The documentation provided gave me a head start to dive into this framework. They have provided several functions that can be used to create and solve various given tasks. They have also provided some functions that can be used for evaluation. You can find some demonstration videos here. [1]

A. Existing Tasks

Here are some of the existing tasks that I found interesting.

1) Inverse Kinematics: In CausalWorld, they have an internal task which performs inverse kinematics to makes the manipulators move to a specific position from an initial position. Given the joint parameters, the position of the arm's end can be calculated using forward kinematics which is fairly simpler compared to the inverse operation which is called inverse kinematics [9]. It is impressive that in addition to many internal tasks available, it also allows us to use inverse kinematics.

Following one of the tutorials, I was able to sample a new goal and intervene on this goal state to generate new goals every time. There is an in-built function which computes joint positions so that it could place the manipulators in the goal position. Please refer to the drive link for a demonstration.

2) Pick and Place: This is a basic task which would ultimately help to build complex towers in a few days. I used a pre-trained agent which performs pick and place operation to move a block into goal position. I kept intervened on the object and kept changing its position. The manipulators were able to handle it sometimes but other times they were not able to pick the block from certain positions due to an obstacle (wall) in the way. Please refer to the drive for a demonstration.

3) Stacking Blocks: This turned out to be a very interesting task of stacking 2 blocks, one on the top of other. This followed an in-built policy called GraspingPolicy which grasps the blocks and places them in a goal position but if I disturb it, as shown in the video, it would not be able to account for those unexpected events by default.

B. Task Creation

The tasks that I created are:

- A variant of stacking with 4 levels.
- A variant of towers task with 5 blocks.

My aim is to train policies which could solve these tasks.

C. Model Training

I trained two models for different tasks from the existing tasks CausalWorld has provided:

- *Towers*: I trained a model with the hyperparameters listed in [1].I trained the model for 0.6 million timesteps and with a maximum episode length of 2500. Fig. 6 shows the loss while training the model. The training was very unstable as the loss kept fluctuating. I reduced the learning rate but still loss kept getting fluctuated. I have trained the model on the towers task provided by them and did not use my own for training.
- *Stacking*: I trained a model with the same hyperparamters. I trained the model for 0.9 million timesteps and with a maximum episode length of 2500. Fig. 5 shows the loss while training the model. The loss did not fluctuate as much as it did for the towers task.

D. Policy Creation

I created two separate policies for towers and stacking tasks using the models I trained.

[1] Drive link: https://drive.google.com/drive/folders/1J4lWqFE6qb3_67wdu2oXKSRpzP_mlRD3?usp=sharing

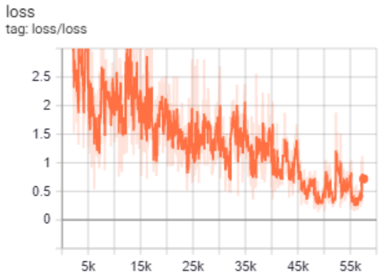


Fig. 5. Loss while training a policy to solve stacking task

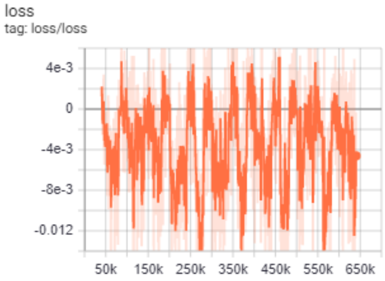


Fig. 6. Loss while training a policy to solve towers task

E. Solving the tasks

I tried to solve the tasks that I created before by modifying the existing towers and stacking tasks. The results can be seen in the results section.

VIII. Evaluation

Models can be trained under three different curricula as explained in [1].

- Curriculum 0: We do not perform any interventions here. The variables in the environment remain constant.
- Curriculum 1: At the beginning of each new episode, a new goal shape is generated i.e, interventions are performed on goal positions and orientation.
- Curriculum 2: At the beginning of each new episode, interventions are performed on all variables.

Curriculum 1 and 2 improve the generalization capabilities of the trained agents. They also defined 12 evaluation protocols which evaluate the generalization capabilities of each agent. Fractional success score is computed at last time step of each episode. Fractional success score is the fractional volumetric overlap of the blocks with the goal shape which ranges between 0 (no overlap) and 1 (full overlap). Each of these protocols are represented by 'P' followed by the protocol number (P0 to P11). They have explained what some of these protocols mean:

- P0: How well the agents can perform the default task.
- P4: How well the agents can generalize when the initial pose changes.
- P5: How well the agents can generalize when the goal pose changes.

A. Stacking Results

I did not find any existing evaluation protocols for evaluating the trained model in the towers task nor there is a way to define the protocols on our own. We will look at the limitations of the framework more closely in the 'Limitations' section of the paper. Fig. 7 and Fig. 8 show the mean fractional success after evaluating the trained model across various given protocols. In the paper, they trained a polciy on the stacking task for 100 million timesteps and the results they got are a bit different that what I got. They also compared the results from PPO with other RL algorithms like Soft Actor-Critic (SAC) and Twin Delayed DDPG (TD3). I have taken Fig. 9 from [1]. The fact that the model that I trained was not able to perform well on the default task but performed better on the other protocols seemed a bit odd.

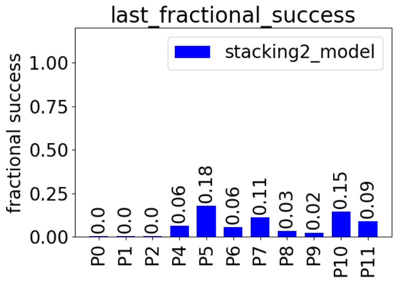


Fig. 7. Bar plot showing the fractional success of the stacking task across different protocols

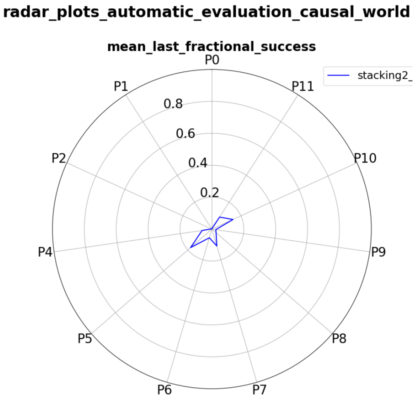


Fig. 8. Radar plot showing the fractional success of the stacking task across different protocols

B. Generalization Capabilities for the Pushing Task

In the CausalWorld Github repository, the authors mentioned that the curriculum learning wrapper cannot be used at the moment. This prevented me from training models in curriculum 1 and 2 modes. I found pretrained models for pushing in curriculum 0 and 1 modes in their repository. I used these for evaluation. Fig. 10 and Fig. 11 show the results from evaluating the Pushing task.

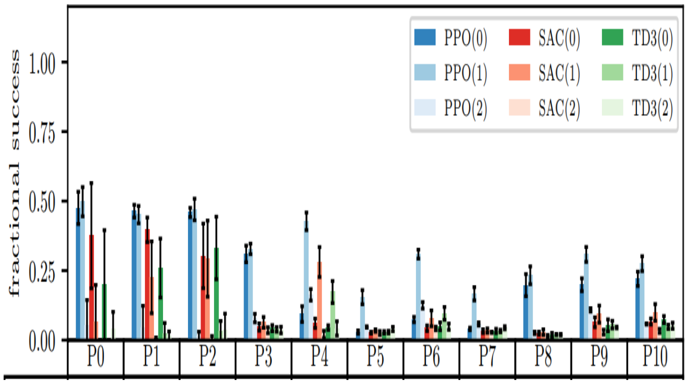


Fig. 9. Evaluation scores for Stacking task from PPO, SAC and TD3 algorithms

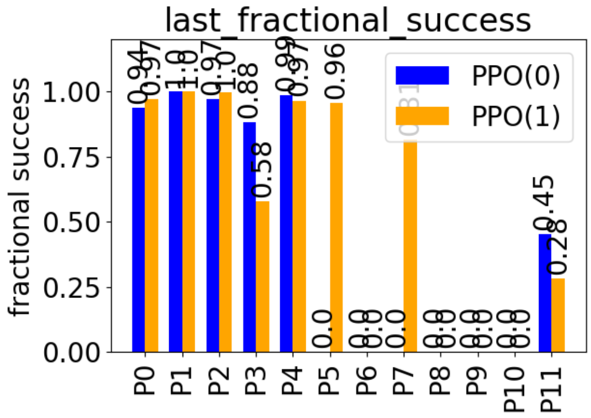


Fig. 10. Bar plot showing the fractional success of the Pushing task across different protocols and two curricula

IX. Limitations

Though I enjoyed working on this project and learnt a lot of new things, there are certain limitations that have prevented me from doing what I wanted to do. Here are some of the drawbacks of the framework.

- There are no existing policies for some of the tasks they have provided which is fine but when I train the new policies for these tasks, it is leading to very unstable training. I have used the hyperparameters suggested by them in the paper/documentation. I tried changing a couple of hyperparameters but it did improve the training loss by much.
- Some interventions might be incompatible with a certain task. One should be very careful while training the model using random inter
- The curriculum learning wrapper they have provided cannot be used at the moment as they mentioned in their documentation and repository which has prevented me from training the models in curriculum 1 or 2 modes. It would be very useful to evaluate trained models in both modes as that would show how well can the models generalize when the goal positions change in the environment.
- Though they have provided various protocols, even their functionality for creating our own evaluation protocols is still not in place. They have mentioned this in their repository. This prevented me from evaluating my trained model across these different protocols.
- Some things were left unexplained and I did not find any other source which would explain these. For example, they did not explain what each one of these protocols actually mean.
- After creating my own policies to solve tasks, the robot was not able to solve the task properly. This happened for Stacking task as well even though the training was smoother for this compared to the towers task. I tried contacting the first author of the paper regarding this issue but did not get any response from him. It seemed they have deleted their discord group as well so there is no way to get help.

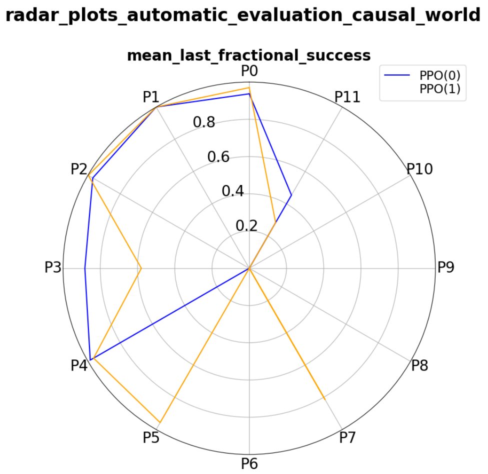


Fig. 11. Radar plot showing the fractional success of the Pushing task across different protocols and two curricula

X. Conclusion and Future Work

Causal Reasoning has always been a crucial cognitive skill essential for human evolution and would leverage the capabilities of robots when this reasoning is inculcated in them. I explored the framework entirely and tested almost all the functionalities that the framework had to offer to assess its capabilities and limitations. I have trained two policies from scratch to solve the towers and stacking tasks. I evaluated the stacking task across different evaluation protocols and was able to produce some results and their interpretation. I found some drawbacks while using the framework and listed them in the paper. Some of these drawbacks prevented me from doing specific tasks that I planned on doing using the framework. Despite its drawbacks, the framework has a wide variety of functions to offer and there is a great scope for improving the framework a lot. Some internet forums dedicated to the framework would help the authors to get proper attention and provide them enough feedback to improve their framework.

I am currently exploring how I could use the concepts of causal reasoning in Reinforcement Learning. One of the ongoing line of works that we are doing in this domain is a multimodal chatbot for teaching students to play Rubik's Cube by using deep reinforcement learning [15]. My role in the project is to make the system more transparent and improve the conversation of the chatbot. Causal Reasoning would make the system more transparent and they can be help accountable. In addition to this, I want to follow-up with other authors again regarding the issues I faced. CausalWorld framework is restricted to a small environment where the number of tasks that you can do are limited. I want to build a better framework in the future which could be used to solve multiple tasks by making use of the interventions to augment the training data which would lead to better generalization.

References

[1] Ahmed, Ossama, et al. ”Causalworld: A robotic manipulation benchmark for causal structure and transfer learning.” arXiv preprint arXiv:2010.04296 (2020).

[2] Wüthrich, Manuel, et al. ”Trifinger: An open-source robot for learning dexterity.” arXiv preprint arXiv:2008.03596 (2020).

[3] Catherine Bernier, ”Human-Robot Interaction: Playing Tower of Hanoi with Robots”. Accessed on: March 15, 2022. Available: https://blog.robotiq.com/bid/66471/Human-Robot-Interaction-Playing-Tower-of-Hanoi-with-Robots

[4] John Aldrich. ”Correlations Genuine and Spurious in Pearson and Yule.” Statist. Sci. 10 (4) 364 - 376, November, 1995. https://doi.org/10.1214/ss/1177009870

[5] S. C. Smith and S. Ramamoorthy, ”Counterfactual Explanation and Causal Inference In Service of Robustness in Robot Control,” 2020 Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob), 2020, pp. 1-8, doi: 10.1109/ICDL-EpiRob48136.2020.9278061.

[6] Hellström, Thomas. ”The relevance of causation in robotics: A review, categorization, and analysis” Paladyn, Journal of Behavioral Robotics, vol. 12, no. 1, 2021, pp. 238-255. https://doi.org/10.1515/pjbr-2021-0017

[7] Levine, Steven James, and Brian Charles Williams. ”Concurrent plan recognition and execution for human-robot teams.” Twenty-Fourth International Conference on Automated Planning and Scheduling. 2014.

[8] G. Katz, D. Huang, T. Hauge, R. Gentili and J. Reggia, ”A Novel Parsimonious Cause-Effect Reasoning Algorithm for Robot Imitation and Plan Recognition,” in IEEE Transactions on Cognitive and Developmental Systems, vol. 10, no. 2, pp. 177-193, June 2018, doi: 10.1109/TCDS.2017.2651643.

[9] D. L. PIEPER, ”THE KINEMATICS OF MANIPULATORS UNDER COMPUTER CONTROL.” Order No. 6914038, Stanford University, Ann Arbor, 1969.

[10] Judea Pearl. 2009. Causality: Models, Reasoning and Inference (2nd. ed.). Cambridge University Press, USA.

[11] Pearl, Judea, and Dana Mackenzie. 2019. The Book of Why. Harlow, England: Penguin Books.

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

[13] OpenAI ”Proximal Policy Optimization”. Accessed on: May 2, 2022. Available: https://openai.com/blog/openai-baselines-ppo/

[14] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W. (2016). Openai gym. ArXiv Preprint ArXiv:1606.01540.

[15] Kausik Lakkaraju, Thahimum Hassan, Vedant Khandelwal, Prathamjeet Singh, Cassidy Bradley, Ronak Shah, Forest Agostinelli, Biplav Srivastava, Dezhi Wu ”ALLURE: A Multi-Modal Guided Environment for Helping Children Learn to Solve a Rubik’s Cube with Automatic Solving and Interactive Explanations” AAAI 2022