Learning the Principle of Least Action with Reinforcement Learning

Zehao Jin[1*], **Joshua Yao-Yu Lin**[2*], **Siao-Fong Li**[3],
[1]Center for Astro, Particle and Planetary Physics (CAP[3]), New York University Abu Dhabi
[2]University of Illinois at Urbana-Champaign
[3]University of Massachusetts, Amherst

Abstract

It is attractive to learn physics via machine learning because physics describes our complicated real-world both elegantly and economically, with simple laws of physics to govern the evolution of complex states. In the case of classical mechanics, nature favors the object to move along the path according to the time integral of the Lagrangian, called the action $\mathcal{S}$. We consider setting the reward/penalty as a function of $\mathcal{S}$, so the agent could learn the physical trajectory of particles in various kinds of environments with reinforcement learning (RL). In this work, we verified the idea by using a Q-Learning based algorithm on learning how light propagates in materials with different refraction indices, and show that the agent could recover the minimal-time path equivalent to the solution obtained by Snell's law or Fermat's Principle. The success sheds light on the possibility of further applications for combining RL and physics.

Introduction

The knowledge of physics, from a pragmatism viewpoint, is a synthesis of mathematical formulas that make predictions based on the input information. For example, Newtonian mechanics (e.g. Newton's three laws of motion) have been very successful in describing the equation of motion in a classical world. There are two variants of Newtonian mechanics, Lagrangian and Hamiltonian mechanics that enable us to solve complex systems, for example, multiple pendula, easily because the complicated constraint force can be described by Lagrangian multiplies.

Several studies have been done using deep neural networks with inductive bias that incorporate the Lagrangian or Hamiltonian mechanism to into cost function predict the motion of objects (Lutter, Ritter, and Peters 2019; Cranmer et al. 2020; Greydanus, Dzamba, and Yosinski 2019; Toth et al. 2019). These works show promising results in terms of predicting more physical trajectories compare with vanilla neural networks with simple cost functions. However, these frameworks could be restrictive because the training set of the system of interest needs to be designed exactly as the testing set, while in reality, the governing dynamics of the task of interest are usually unknown, or only partial information is revealed. Here we proposed that we could use reinforcement learning to learn the trajectory of a physical system. The least action principle indicates that the physical action $\mathcal{S}$, an time integration of a Lagrangian $\int dtL$, will be minimized by the classical physics path.

The fact raised the idea to use the least action principle as the basis of a learning algorithm, rather than Lagrangian or Hamiltonian, for a physics model related to an optimized path. Because the physical action $\mathcal{S}$ is able to be interpreted as an intrinsic property of the environment, it shows an opportunity to use a non-supervised learning algorithm. Therefore, we decided to implement Reinforcement Learning (RL) which shows promising progress in various applications, especially when applied to games (Mnih et al. 2013; Silver et al. 2016), robotic (Kalashnikov et al. 2018) and scientific discovery or design (Halverson, Nelson, and Ruehle 2019; Garnier et al. 2019; Popova, Isayev, and Tropsha 2018; Denil et al. 2016).

We used the refraction of light combined with Q-Learning as an example to demonstrate the concept (Watkins 1989). Our agent will then search along with the interface (RL action) for the location of incident points (RL states) at each interface that gives the light path of the shortest time. That is, the agent will always get some reward in each round depending on its choice of incident points in each material.

We noticed the word *action* is used in a different way in physics and RL. In this work we use "physical action" $\mathcal{S}$ to represent the physical quantity, not to be confused with the RL action a.

The Q-learning algorithm with a reward function of Physical Action $\mathcal{S}$

We set up our environment with starting point A and destination B, as shown in Figure 1, and then letting light rays (agent) travel in different materials (environment). For simplicity, light rays are restricted to move in a straight line in the same material. Each light path is evaluated with a reward function dependent on $\mathcal{S}$, in each round is just straight line segments connecting starting point A, incident points at different interfaces, and final point B. Our agent will then search along with the interface (RL action) for the location of incident points (RL states) at each interface that gives the

[*]equal contribution

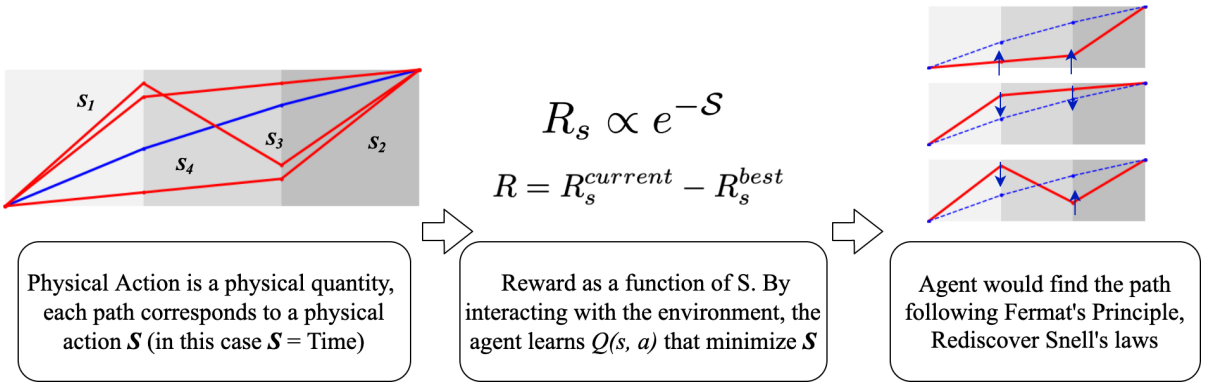

Figure 1: A cartoon summarizing how we connect the principle of least action and reinforcement learning in the case of light refraction.

light path of the shortest time. That is, the agent will always get some reward in each round depending on its choice of incident points in each material.

Physical Action from Fermat's Principle

Fermat's principle states that light rays travels between two points along the path that requires the least time. In order to determine the time the light ray spends between two points A and B, we could integrate the time dt at every instance, which would be distance divided by its ray's velocity in the media,

$$\mathcal{T} = \int_A^B dt = \int_A^B \frac{ds}{v_r}, \tag{1}$$

where v_r represents the speed of light in the medium. Furthermore, given the speed of light in vacuum $c = 1$, it is generalized to optical path length that would have the form of (physical action)(Chaves 2017), namely

$$\mathcal{S} = \mathcal{T} = \int_A^B n_r ds. \tag{2}$$

Once we have the physical action, we could choose a reward as a function of the physical action, in this case, a function of time as the reward.

Experiment

Q-Learning (Watkins and Dayan 1992) is a basic RL algorithm that builds a Q-table that keeps record of Q-values of all available actions at all possible states. The Q-value is updated based on the reward the agent received by making a particular action at a particular state. At any state, there is an ϵ chance for the agent to make an action that has the highest Q-value, otherwise the agent will take a random action for more exploration. In this work we adopted the greedy factor $\epsilon = 0.9$, learning rate α = 0.001, discount factor γ = 0.9. An overview of our experiment is described in the table and visualized in Figure 2.

The time T the light ray costs through a distance l in a material with index of refraction n_r is $T = l \cdot n_r$. As light travels through multiple different materials, the total time it takes is just the sum of time took in each material, $T = \Sigma_i T_i$, and $T_i = l_i \cdot n_{r,i}$.

We constructed a three-layer, 50×150 grid environment that consists of three 50×50 grid materials of air, water and glass from left to right. The given endpoints for our RL agent is from bottom left corner(A) to top right corner(B).

The RL state of our agent is State=(y_1-coordinate of air-water interface, y_2-coordinate of water-glass interface). At the beginning of each training episode, our agent starts from initial state $s_{ini}(y_1, y_2) = (0, 0)$, and the theoretical least-time light path is state $s_{theo}(y_1.y_2) = (21, 37)$. Each round our agent moves up/down one unit along one of the two interfaces. That is, for each round, the one of the four RL actions $a = \{y1\uparrow, y1\downarrow, y2\uparrow, y2\downarrow\}$ is taken, where the arrow means moving along the direction for one unit.

We defined a R score, R_s as

$$R_s = Ne^{-\mathcal{S}} = Ne^{-T} \tag{3}$$

where T is the total time it takes for our agent to travel between two endpoints, and N is just an arbitrary scaling factor. The $e^{-\mathcal{S}}$ form is taken from the Euclidean path integral formalism (Hall 2013). The reward our agent receives for each round is the difference between R_s for this round and the best R_s achieved so far in this episode.

$$R = R_s^{\text{current}} - R_s^{\text{best}} \tag{4}$$

The reward is defined this way so that the agent is would get reward if the current path is better than the path it has explored, and vice versa. We find that in this particular environment setting, this definition of reward and R_s help the agent find a global maximum faster.

Our agent is trained for 100 episodes, and during each episode our agent moves 300 rounds. The training result for each episode can be visualized in Figure 3, and as an example, Figure 2 visualizes training episode #90. Our agent is able to find the path that takes the least time.

To generalize the problem a bit,the agent is also trained under indices of refraction other than $(n_1, n_2, n_3) = (1, 1.3, 1.6)$, and with initial states other than $s_{ini}(y_1, y_2) = (0, 0)$. The agent can still successfully find the correct path. One of those trials are shown in Figure 4.

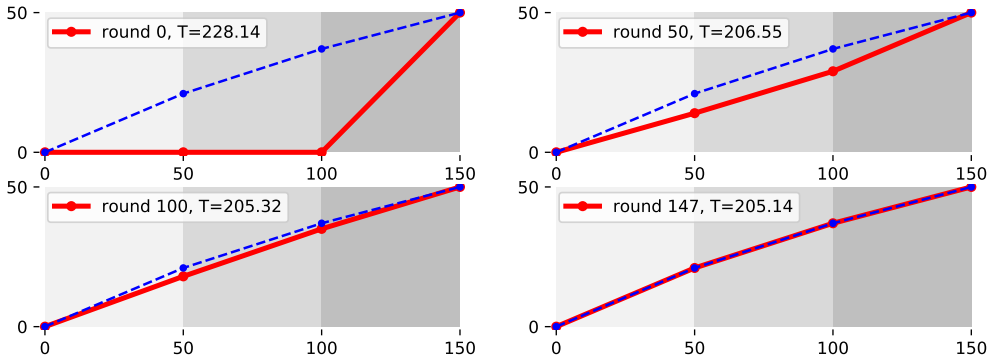

Figure 2: The evolution of learning during a single training episode. The environment of air($n_{air} = 1$), water($n_{water} = 1.3$), and glass($n_{glass} = 1.6$) from left to right. Our agent is asked to travel from bottom left corner to top right corner. The red lines are the light path our agent chose and the blue dotted line is the theoretical least-time light path.

Q-Learning and Environment General parameters		
RL State	s	(y_1, y_2), with $0 \leq y_1 \leq 50$ and $0 \leq y_2 \leq 50$
RL Action	a	$y1\uparrow, y1\downarrow, y2\uparrow, y2\downarrow$
RL Reward	R	$R = R_s^{\text{current}} - R_s^{\text{best}}$, $R_s = Ne^{-T}$
Total Time	T	$T = \Sigma_i T_i, \quad T_i = l_i \cdot n_{r,i}$
Greedy factor	ϵ	$\epsilon = 0.9$
Learning rate	α	$\alpha = 0.001$
Discount factor	γ	$\gamma = 0.9$
Parameters specifically for the case in Figure 2		
Index of refraction	n_i	left to right: $n_{air} = 1$, $n_{water} = 1.3$, $n_{glass} = 1.6$
Path endpoints	A, B	$A(x, y) = (0, 0), B(x, y) = (150, 50)$
Initial state	s_{ini}	$s_{ini}(y_1, y_2) = (0, 0)$
Snell's Law prediction	s_{theo}	$s_{theo}(y_1, y_2) = (21, 37)$
Total training episode		100
Rounds in each episode		300

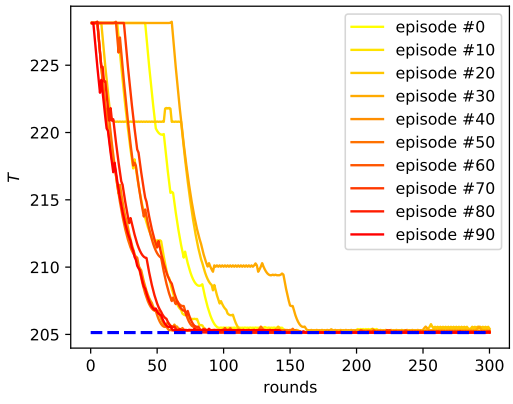

Figure 3: Time our agent took to travel between the given endpoint for each training round during 10 different training episodes. The blue dotted line denotes the theoretical shortest time. The episode #90 is taken as an example in Figure 2.

Discussion

We propose a new physics-learning framework based on RL that utilized the concept of the least action principle, which indicates that the physical action $\mathcal{S}$ should be minimum, as its reward function. To demonstrate the idea, we used Q-learning on the problem of refraction of light among materials. The agent successfully learns the least-time path with the reward function $e^{-\mathcal{S}}$, although we highly restricted the phase space of actions, only the movements of y coordinates of interfaces. The restriction is possible to be relaxed when there are more computation resources or more suitable algorithms in the future.

Here we want to raise an important question: could we claim that our work is prior to the knowledge of physics? There will be a positive answer, if we consider our work from the path-learning perspective. Paths are not supervised and constrained by any special condition. The whole learning is based on the minimization of $\mathcal{S}$, the Fermat's principle, a very important knowledge of physics embedded as an intrinsic property of the reward function. Therefore, the answer depends on the definition of the physics knowledge.

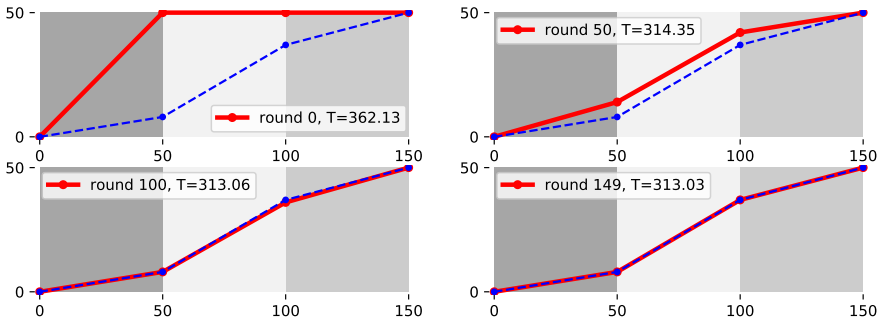

Figure 4: The environment of refraction index $n_1, n_2, n_3 = (3, 1, 2)$ from left to right. The search begins from initial state $s_{ini}(y_1, y_2) = (50, 50)$. The red lines are the light path our agent chose during episode #90, and the blue dotted line is the theoretical least-time light path.

Nevertheless, our work still clearly shows that the path is possible to be decided by physical action $\mathcal{S}$. Vice versa, it is also possible to learn physical action $\mathcal{S}$ from path under the least action principle, too. We will explore the possibility in the future.

While for classical physics, most of the path-finding problem could be treated as a traditional optimization problem, we noticed that our RL framework provides a potential link to learn quantum mechanics. In the quantum world, which non-optimal path could also contribute to the final state, and our RL framework could provide a way to evaluate the "value" for the non-optimal path that could not be capture easily by traditional optimization. The mathematical tool to describe Quantum mechanics is a wave function of space-time. Compare the wave function to the coordinates in classical physics, the amount of data is incredibly huge, the whole space v.s. one point. However, physicist cleverly observed that the quantum mechanics is able to be described by the physical action $\mathcal{S}$ from the all possible paths, which is called as path integral (Dirac 1981), and more importantly, the *exploration* and *exploitation* nature of RL approach can be used to pick up the truly important paths, especially with deep reinforcement learning.

Conclusion

We demonstrate that RL can be applied to path finding in a physical system by purely interacting with the environment and getting the path of least action without knowing the ground truth. We believe that with more computational resources and advanced RL algorithm, RL can be applied to learn complex physics models based on the physical action $\mathcal{S}$. We also noticed that the potential of *exploration* and *exploitation* nature of RL approach is actually similar to the spirit of path integrals in quantum mechanics(Dirac 1981), where all of the conceivable (non-optimized) path could also contribute so the all possible path between two points needs to be explored and evaluated. We plan to investigate these ideas in our future work.

References

Chaves, J. 2017. *Introduction to nonimaging optics*. Taylor & Francis.

Cranmer, M.; Greydanus, S.; Hoyer, S.; Battaglia, P.; Spergel, D.; and Ho, S. 2020. Lagrangian neural networks. *arXiv preprint arXiv:2003.04630* .

Denil, M.; Agrawal, P.; Kulkarni, T. D.; Erez, T.; Battaglia, P.; and De Freitas, N. 2016. Learning to perform physics experiments via deep reinforcement learning. *arXiv preprint arXiv:1611.01843* .

Dirac, P. A. M. 1981. *The principles of quantum mechanics*. 27. Oxford university press.

Garnier, P.; Viquerat, J.; Rabault, J.; Larcher, A.; Kuhnle, A.; and Hachem, E. 2019. A review on deep reinforcement learning for fluid mechanics. *arXiv preprint arXiv:1908.04127* .

Greydanus, S.; Dzamba, M.; and Yosinski, J. 2019. Hamiltonian neural networks. In *Advances in Neural Information Processing Systems*, 15379–15389.

Hall, B. C. 2013. *Quantum theory for mathematicians*. Springer.

Halverson, J.; Nelson, B.; and Ruehle, F. 2019. Branes with brains: exploring string vacua with deep reinforcement learning. *Journal of High Energy Physics* 2019(6): 3.

Kalashnikov, D.; Irpan, A.; Pastor, P.; Ibarz, J.; Herzog, A.; Jang, E.; Quillen, D.; Holly, E.; Kalakrishnan, M.; Vanhoucke, V.; et al. 2018. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293* .

Lutter, M.; Ritter, C.; and Peters, J. 2019. Deep lagrangian networks: Using physics as model prior for deep learning. *arXiv preprint arXiv:1907.04490* .

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* .

Popova, M.; Isayev, O.; and Tropsha, A. 2018. Deep reinforcement learning for de novo drug design. *Science advances* 4(7): eaap7885.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature* 529(7587): 484–489.

Toth, P.; Rezende, D. J.; Jaegle, A.; Racanière, S.; Botev, A.; and Higgins, I. 2019. Hamiltonian generative networks. *arXiv preprint arXiv:1909.13789* .

Watkins, C.; and Dayan, P. 1992. Technical Note: Q-Learning. *Machine Learning* 8: 279–292. doi:10.1007/BF00992698.

Watkins, C. J. C. H. 1989. Learning from delayed rewards .