Perceiving Physical Equation by Observing Visual Scenarios

Siyu Huang[*]
Zhejiang University
siyuhuang@zju.edu.cn

Zhi-Qi Cheng[*]
Southwest Jiaotong University
zhiqicheng@gmail.com

Xi Li
Zhejiang University
xilizju@zju.edu.cn

Xiao Wu
Southwest Jiaotong University
wuxiaohk@gmail.com

Zhongfei (Mark) Zhang
Zhejiang University
zhongfei@zju.edu.cn

Alexander Hauptmann
Carnegie Mellon University
alex@cs.cmu.edu

Abstract

Inferring universal laws of the environment is an important ability of human intelligence as well as a symbol of general AI. In this paper, we take a step toward this goal such that we introduce a new challenging problem of inferring invariant physical equation from visual scenarios. For instance, teaching a machine to automatically derive the gravitational acceleration formula by watching a free-falling object. To tackle this challenge, we present a novel pipeline comprised of an Observer Engine and a Physicist Engine by respectively imitating the actions of an observer and a physicist in the real world. Generally, the Observer Engine watches the visual scenarios and then extracting the physical properties of objects. The Physicist Engine analyses these data and then summarizing the inherent laws of object dynamics. Specifically, the learned laws are expressed by mathematical equations such that they are more interpretable than the results given by common probabilistic models. Experiments on synthetic videos have shown that our pipeline is able to discover physical equations on various physical worlds with different visual appearances.

1 Introduction

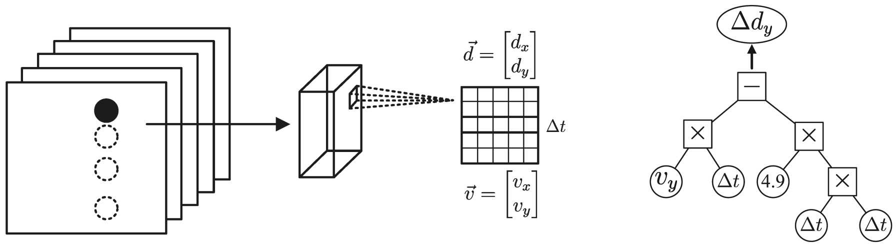
Inference is one of the most basic and significant aspects of human intelligence [1] as well as AI [2]. As a high-level aspect of inference, the induction of universal laws from observations of our world is both the core basis and the goal of the scientific research. For example, Sir Isaac Newton saw an apple falling down and then was inspired to discover the law of gravitation. However, for a computing machine, the induction of laws based on visual observations is still a very challenging and open problem, and has been rarely explored by the existing literature until today.

In this paper, we introduce a new problem that we attempt to teach machine to automatically derive mathematical expressions of object dynamics from videos of a physical world. In contrast to the most recent approaches [3–5] which explores to learn object mechanical behaviors by the black box of deep neural networks, we aim at explicitly presenting the symbolic expressions of latent physical laws, leading to a more interpretable model and more visualizable results. A pioneer work [6] learns to derive mathematical equations from the data of physical experiments. While in this work, we propose to learn mathematical expressions directly from complicated videos.

Toward this goal, we propose a novel pipeline comprised of an *Observer Engine* and a *Physicist Engine*. The Observer Engine acts like an observer that watches the videos of a physical scenario

[*]Equal contributions. Work done when Siyu Huang and Zhi-Qi Cheng visited Carnegie Mellon University.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.



(a) Observer Engine (b) Physicist Engine

Figure 1: Our model is comprised of (a) the Observer Engine and (b) the Physicist Engine. At left, a video depicts that an object is in free-falling. The Observer Engine uses deep neural networks to extract the physical properties of the object. The Physicist Engine learns a mathematical expression of the object dynamics by evolving a syntax tree based on the property variables.

and extracts the physical properties of objects in that scenario. Then the Physicist Engine imitates a physicist that summarizes the observed data and finally derives the mathematical equations.

In the experiments, we evaluate our pipeline on synthetic videos of multiple physical scenarios, showing that it is able to learn precise mathematical equations on these physical worlds with diverse visual appearances. We also explore several variants of models for the Observer Engine and the Physicist Engine respectively, so as to quantitatively establish baselines for relevant research in the future.

Our contributions are three-fold. First, we introduce a new problem of learning mathematical equations of object dynamics from videos, taking a step toward the automatic induction of universal laws for general AI. Second, we propose a novel pipeline to tackle this challenging problem. Third, empirical studies demonstrate the effectiveness of our approach on several synthetic physical scenarios.

2 Model

Our model learns to infer the inherent mathematical equation from video frames of a physical system. It consists of an *Observer Engine* and a *Physicist Engine*.

Observer Engine The Observer Engine acts like an observer that watches the videos of a physical world, and at the same time records the physical-property variables. As illustrated in Fig. 1(a), it captures the physical properties of the kinetic objects and the environment in videos. In this work, we use the Faster-RCNN [7] model to detect an object and localize its position $\vec{d}$ according to coordinates of the bounding-boxes. In order to get a more precise object position, we employ a two-stage approach to refine the position on coarse-to-fine spatial scales. Specifically, a Faster-RCNN detector is applied on an image to get a coarse window of an object, then another Faster-RCNN detector is applied on the window to get a fine bounding-box. The two-stage approach ensures a precise object localization and a speed up of the detection procedure. The velocity $\vec{v}$ of an object is computed by $\vec{v} = \Delta\vec{d}/\Delta t$, where Δt is the time interval between two video frames. Observation data $\vec{d}$, $\vec{v}$, and Δt are fed to the Physicist Engine serving as the independent variables.

Physicist Engine The Physicist Engine acts like a physicist that infers the equation based on the observations given by the Observer Engine. It takes a set of objects' physical properties (output from the visual engine applied to a series of videos) as input. It outputs the equation between displacement $\Delta\vec{d}$ and the independent variables. In this work, we adopt symbolic regression with genetic programming (GP) [8, 9] for the inference of mathematical equation, implemented based on GPlearn Toolkit[2].

[2]http://gplearn.readthedocs.io/en/stable/index.html

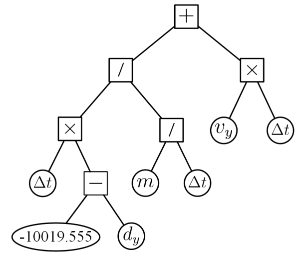
Physical Scenario	Learned Δd_x	Learned Δd_y
 **Spring** (`http://bit.ly/2KuAfnj`)	0.0 0	 $v_y\Delta t - (d_y + 10019.555)/m \cdot \Delta t^2$

Figure 2: An example of the synthetic physical scenario and the learned equations.

As illustrated in Fig. 1(b), the formula is represented as a syntax tree. The variables, denoted as the round nodes, are leaves of the tree. The mathematical operations, denoted as the square nodes, connect the independent variables. Our goal is to find the best formula consisting of arbitrary independent variables and mathematical operations to minimize the mean absolute error (MAE) corresponding to the given target. At the very beginning, a population of formulas is randomly initialized. In an evolutionary manner, GP evolves the fittest ones of every generation until convergence. More details of GP are discussed in our Appendix A.3.

3 Experiments

Qualitative example We conduct experiments on several different types of physical scenarios. In each scenario, there is an object obeying the basic dynamic equation as

$$\Delta\vec{d} = \vec{v}\Delta t + \frac{1}{2}\vec{a}\Delta t^2 \tag{1}$$

Fig. 2 shows an example of the synthetic physical scenario. The object is connected to a horizontal wall with a visible spring obeying Hooke's law. The accelerated velocity $\vec{a}_{\text{spring}}$ is

$$\vec{a}_{\text{spring}} = \begin{bmatrix} 0 \\ -k \cdot (d_y - D - X)/m \end{bmatrix} \tag{2}$$

The Hooke's constant k, attachment point y-coordinate D, and equilibrium distance X are constants. In this experiment, we set $k = 2$, $D = -15000$, and $X = 5000$.

As shown in the left part of Fig. 2, the Observer Engine detects the bounding boxes (green) of objects, providing precise object positions to the Physicist Engine. At the right part of Fig. 2, we show the mathematical equations and the syntax trees learned by the Physicist Engine. The Physicist Engine presents precise dynamic equations, even though the dynamic equation of Δd_y is somewhat complex. Not only the symbolic relationships are correctly learned, the physical constants in mathematical equations are also accurately estimated by our method. More examples of synthetic physical scenarios are shown in the Appendix B.1.

Pipeline study Table 1 shows the ablation study of different baseline methods for our pipeline. The baseline methods of two engines are pairwise combined to be evaluated in all the physical scenarios. For the Physicist Engine, we study several typical regression methods for a comparison with our used symbolic regression method, including (1) linear regression, (2) ridge regression, (3) decision tree, and (4) random forest. For the Observer Engine, we study two baseline methods for a comparison with our used Two-stage Detector, including (a) Single Detector which is a basic Faster R-CNN [7] model and (b) Detection + Segmentation.

We use R^2 coefficient score as the metric to evaluate the fitting goodness in this study. It is interesting that when working with Single Detector or Detection + Segmentation, sometimes the methods of Physicist Engine perform better than the ground-truth equation. It is mainly because these methods

Table 1: Ablation study of our pipeline (R^2 score). Baseline methods of the Observer Engine and the Physicist Engine are row-wise listed and column-wise listed respectively. LR: linear regression; RR: ridge regression; DT: decision tree; RF: random forest; SR: symbolic regression; GT: ground-truth equation.

	LR	RR	DT	RF	SR	GT
Single Detector	0.917	0.908	0.812	0.932	0.926	0.904
Detection + Segmentation	0.958	0.958	0.832	0.922	0.954	0.954
Two-stage Detector	0.945	0.944	0.960	0.983	**1.000**	1.000
Ground-truth Position	0.945	0.944	0.970	0.984	1.000	1.000

eliminate some position errors in fitting. We observe that our pipeline (a combination of Two-stage Detector and SR) gets an 1.000 R^2 score, as it successfully identifies all of the dynamic equations as well as accurately estimates the constants. Comparing Two-stage Detector with Ground-truth Position and comparing SR with GT, both of them show performances close to the ground-truth, indicating that they have good compatibilities with different methods of the other engine.

4 Discussion

We have introduced a new problem of deriving mathematical equations from physical scenarios, taking a step toward the goal of reasoning about universal laws from a complex environment. We have presented a pipeline including an Observer Engine and a Physicist Engine to tackle this problem for the first time. In the experiments, we have shown that our pipeline is able to perceive dynamic equations on synthetic physical scenarios with noisy visual appearances. Ablation studies conducted on combinations of baselines further demonstrate the effectiveness of our pipeline. By combining deep learning, symbolic learning, and evolutionary algorithm, our approach shows the potential of a hybrid machine learning system for AI reasoning.

In the future, an important work is to demonstrate the proposed pipeline in real-world scenarios which may have more unknown noise than the synthetic data. It will also be important to develop techniques to handle the multi-object physical system [3–5], in which there are interactions between objects other than the dynamics of a single object, such that we need to learn a composite set of dynamic laws in a scenario.

References

[1] Joshua B Tenenbaum, Charles Kemp, Thomas L Griffiths, and Noah D Goodman. How to grow a mind: Statistics, structure, and abstraction. *science*, 331(6022):1279–1285, 2011.

[2] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017.

[3] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. In *NIPS*, pages 4502–4510, 2016.

[4] Nicholas Watters, Daniel Zoran, Theophane Weber, Peter Battaglia, Razvan Pascanu, and Andrea Tacchetti. Visual interaction networks: Learning a physics simulator from video. In *NIPS*, pages 4542–4550, 2017.

[5] Jiajun Wu, Erika Lu, Pushmeet Kohli, Bill Freeman, and Josh Tenenbaum. Learning to see physics via visual de-animation. In *NIPS*, pages 152–163, 2017.

[6] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.

[7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[8] Douglas Adriano Augusto and Helio JC Barbosa. Symbolic regression via genetic programming. In *Proceedings. Sixth Brazilian Symposium on Neural Networks*, pages 173–178, 2000.

[9] Orazio Giustolisi and Dragan A Savic. A symbolic data-driven technique based on evolutionary polynomial regression. *Journal of Hydroinformatics*, 8(3):207–222, 2006.