A Compositional Object-Based Approach to Learning Physical Dynamics

Michael Chang[*], **Tomer Ullman**[**], **Antonio Torralba**[*], and **Joshua B. Tenenbaum**[**]
[*]Department of Electrical Engineering and Computer Science, MIT
[**]Department of Brain and Cognitive Sciences, MIT
{mbchang,tomeru,torralba,jbt}@mit.edu

Abstract

This paper presents the Neural Physics Engine (NPE), an object-based neural network architecture for learning predictive models of intuitive physics. The NPE draws on the strengths of both symbolic and neural approaches: like a symbolic physics engine, it is endowed with generic notions of objects and their interactions, but as a neural network it can also be trained via stochastic gradient descent to adapt to specific object properties and dynamics of different worlds. We evaluate the efficacy of our approach on simple rigid body dynamics in two-dimensional worlds of bouncing balls. By comparing to a less structured architecture, we show that the NPE's compositional representation of the causal structure in physical interactions improves its ability to predict movement, generalize to different numbers of objects, and infer latent properties of objects such as mass.

1 Introduction

A sense of intuitive physics can be seen as a program [7] that takes in input provided by a physical scene and the past states of objects and then outputs the future states and physical properties of relevant objects for a given task. At least two general approaches have emerged in the search for such a program that captures common-sense physical reasoning. The top-down approach [3, 16, 17] formulates the problem as inference over the parameters of a symbolic physics engine, while the bottom-up approach [1, 5, 8–11, 13] learns to directly map physical observations to motion prediction or physical judgments. A program under the top-down approach can express and generalize across any scenario supported by the entities and operators in its description language. However, it may be brittle under scenarios not supported by its description language, and adapting to these new scenarios requires modifying the code or generating new code for the physics engine itself. In contrast, the same model architecture and learning algorithm under gradient-based bottom-up approaches can be applied to new scenarios without requiring the physical dynamics of the scenario to be pre-specified. However, such models require extensive amounts of data, and oftentimes transferring knowledge to new scenes requires retraining, even in cases that seem trivial to human reasoning.

This paper takes a step toward bridging this gap between expressivity and adaptability by proposing a model that combines rough symbolic structure with gradient-based learning. We present the Neural Physics Engine (NPE), a predictive model of physical dynamics. It exhibits several strong inductive biases that are explicitly present in symbolic physics engines, such as a notion of objects and object interactions. It is also end-to-end differentiable and thus is also flexible to tailor itself to the specific object properties and dynamics of a given world through training. This approach – starting with a general sketch of a program and filling in the specifics – is similar to ideas presented by [12, 14]. The NPE's general sketch is the structure of its architecture, and it extends and enriches this sketch to model the specifics of a particular scene by training on observed trajectories from that scene.

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

2 Neural Physics Engine

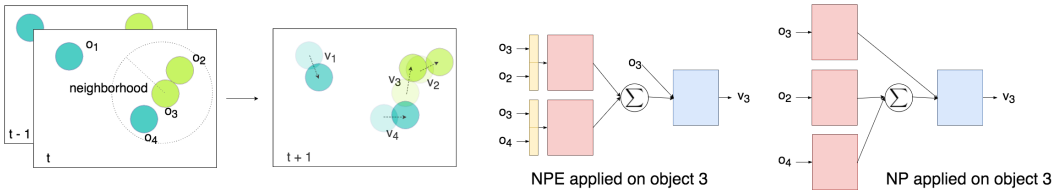

Figure 1: **Scenario and Models** This figure compares the NPE and the NP architectures in predicting the velocity of object 3 for an example scenario [**left**] of two heavy balls (cyan) and two light balls (yellow-green). Objects 2 and 4 are in object 3's neighborhood, so object 1 is ignored. [**middle**]: The NPE encoder consists of a pairwise layer (yellow) of 25 hidden units and a five-layer feedforward network (red) of 50 hidden units per layer each with ReLU activations. Here, it conditions on object 3 as the focus object by encoding its state with object 2 and with object 4 at $[t-1, t]$. The encoding weights are shared across all object pairs. The decoder (blue) takes the sum of the pairwise "effects" and object 3's past state as input, and predicts object 3's velocity for $t+1$. The decoder is a five-layer network with 50 hidden units per layer and ReLU activations after all but the last layer. [**right**]: The NP encoder is the same as the NPE encoder, but without the pairwise layer. Its weights are shared across all objects. The NP decoder is the same as the NPE decoder. Note that incorporating information for modeling object interactions only happens after the NP encoding step.

Jointly predicting the future states of all objects in a scene becomes unscalable as the number of objects grows large. We make two key observations to reduce the complexity of this problem. First, because physical laws do not change across inertial frames, it suffices to separately predict the future state of each object conditioned on the past states of itself and the other objects in its local neighborhood, similar to [5]. Second, because physics is Markovian, this prediction need only be for the immediate next timestep. This spatiotemporal factorization of the scene is the basis of the NPE.

Letting a particular object be the *focus* object f and all other objects in the scene be *context* objects c_k, the NPE models the focus object's velocity $v_{t+1}^{(f)}$ as a composition of the pairwise interactions between itself and other neighboring context objects in the scene during time $t-1$ and t. This input is represented as pairs of object state vectors $(o^{(f)}, o^{(c_1)})_{[t-1,t]}, (o^{(f)}, o^{(c_2)})_{[t-1,t]}, \ldots$. A state vector comprises extrinsic properties (position, velocity, orientation, angular velocity), intrinsic properties (mass, object type, object size), and global properties (gravitational, frictional, and pairwise forces). The NPE also predicts angular velocity along with velocity, but for the experiments in this paper we always set angular velocity, as well as gravity, friction, and pairwise forces, to zero. As shown in Figure 1, the NPE is a composition of an encoder function f_{enc} that summarizes the interaction of a single object pair and a decoder function that takes the sum of encodings of all pairs to predict $v_{t+1}^{(f)}$.

How f_{enc} and f_{dec} are composed emulates the high-level formulation of many symbolic physics engines. We provide a loose interpretation of the encoder output $e^{(f,c)}$ as the *effect* of object c on object f, and require that these effects are additive as forces are, allowing the NPE to scale naturally to different numbers of neighboring context objects. These inductive biases have the effect of strongly constraining the space of possible programs of predictive models that the NPE can learn, focusing on compositional programs that reflect pairwise causal structure in object interactions.

To highlight the advantages of the NPE structure, we compare the NPE to a baseline that is very similar, but lacks the pairwise layer of the encoder. This No-Pairwise (NP) baseline is a Markovian variant of the Social LSTM presented by [2]; it sums the encodings of context objects after encoding each object independently. We compare with this subtly different model to highlight the usefulness of a pairwise factorization of the underlying physics of a scene.

3 Evaluation

Using the matter-js physics engine [4], we evaluate the NPE on worlds of bouncing balls. These worlds exhibit self-evident dynamics and support a wide set of scenarios that reflect everyday physics. Bouncing balls have also been of interest in cognitive science to study causality and counterfactual reasoning, as in [6]. We trained on 3-timestep windows in trajectories of 60 timesteps (10 timesteps $\approx$

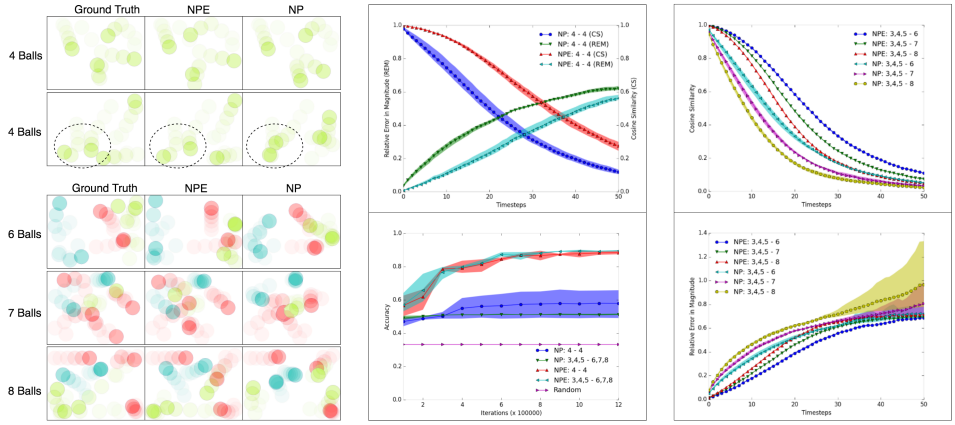

Figure 2: **Results** [**left panel**]: The NPE more successfully resolves collisions than the NP (circled). For generalization tasks with variable mass (cyan: mass = 25, red: mass = 5, yellow-green: mass = 1), the NPE adheres closer to the ground truth whereas the NP causes balls to overlap. [**top-middle**] Cosine similarity (CS) and relative error in magnitude (REM) for the predicted velocity for the prediction task (train on 4 balls, test on 4 balls). [**right panel**] Cosine similarity and relative error in magnitude for the predicted velocity for the generalization task (train on 3, 4, 5 balls, test on 6, 7, 8 balls). [**bottom-middle**] The NPE performs significantly better than the NP in inferring mass. Notably, the NPE performs similarly well whether in a world it has seen before or in a world with a number of objects it hasn't trained on, further showcasing its strong generalization capabilities.

1 second) using rmsprop [15] with a Euclidean loss. Experimental results on test data are summarized in Figure 2. Plots show results over three independent runs with different random seeds.

Prediction In the prediction task, the models are trained and evaluated on worlds with four balls of equal mass. Figure 2 (top-left) illustrates that compared to the NP, the NPE more effectively learns simple Netwonian concepts such as inertial movement and collisions with other objects and the world boundaries. We visualize the cosine similarity as well as the relative error in magnitude between the predicted and ground truth velocity over 50 timesteps of simulation (Figure 2, top-middle). Both the NPE and the NP take timesteps 1 and 2 as initial input, and then use previous predictions as input for future predictions. Because these worlds are chaotic systems, both the NPE and the NP diverge from the ground truth with time, but the NPE's error decays less and more slowly than the NP's.

Strong Generalization We test whether learned knowledge of these simple physics concepts can be transferred to worlds with a number of objects previously unseen. We train on worlds with 3, 4, and 5 balls and test on more complex worlds with 6, 7, and 8 balls, all with equal mass. As shown in Figure 2 (bottom-left), the NPE exhibits much cleaner extrapolation to worlds with more objects. Its predictions are also more consistent and have less error than those of the NP, whose predictions begin to diverge wildly towards the end of 50 timesteps of simulation (2, right panel). The NPE's performance of this generalization task suggests that its architectural inductive biases are useful for generalizing knowledge learned in Markovian domains with causal structure in object interactions. The next paragraph illustrates another aspect of the NPE's strong generalization capability.

Mass Inference We show that the NPE infers latent properties such as mass. This is motivated by experiments in [3], which uses a probabilistic physics simulation engine, the Intuitive Physics Engine (IPE), to infer various properties of a scene configuration. Whereas the rules of the IPE for modeling physical dynamics were manually pre-specified, the NPE learns these rules from observation. We train on the same worlds used in both the Prediction and Generalization tasks, but we uniformly sampled the mass for each ball from the log-spaced set $\{1, 5, 25\}$. For evaluation, we select scenarios exhibiting collisions with the focus object, fix the masses of all objects except that of the focus object, and score the NPE's prediction under all possible mass hypotheses for the focus object. The prediction is scored against the ground-truth under the same Euclidean loss used in training. The mass hypothesis whose prediction yielded the lowest error is the NPE's maximum likelihood estimate of the mass for the focus object. A random model would guess the correct mass 33% with accuracy.

4 Discussion

We have demonstrated a compositional object-based approach to learning physical dynamics in worlds of bouncing balls in several tasks ranging in complexity. Further work includes generalization to unseen object types and physical laws such as in worlds with immovable obstacles and stacked block towers. Because the NPE is differentiable, we expect that by backpropagating prediction error to its input, it may be able to infer the positions of "invisible" objects, whose effects are felt but whose position is unknown. Our results invite questions on how much prior information and structure should and could be given to bottom-up neural networks, and what can be learned without inducing such structure. It would be interesting to explore how similar models to the NPE can be used as subprograms that can be called by parent programs to evolve entity states through time for applications in areas such as model-based planning and model-based reinforcement learning.

References

[1] P. Agrawal, A. Nair, P. Abbeel, J. Malik, and S. Levine. Learning to poke by poking: Experiential learning of intuitive physics. *arXiv preprint arXiv:1606.07419*, 2016.

[2] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces.

[3] P. W. Battaglia, J. B. Hamrick, and J. B. Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110(45):18327–18332, 2013.

[4] L. Brummitt. http://brm.io/matter-js. URL `http://brm.io/matter-js`.

[5] K. Fragkiadaki, P. Agrawal, S. Levine, and J. Malik. Learning visual predictive models of physics for playing billiards. *arXiv preprint arXiv:1511.07404*, 2015.

[6] T. Gerstenberg, N. Goodman, D. A. Lagnado, and J. B. Tenenbaum. Noisy newtons: Unifying process and dependency accounts of causal attribution. In *In proceedings of the 34th*. Citeseer, 2012.

[7] N. D. Goodman and J. B. Tenenbaum. Probabilistic models of cognition, 2016. URL `http://probmods.org`.

[8] A. Lerer, S. Gross, R. Fergus, and J. Malik. Learning physical intuition of block towers by example. *arXiv preprint arXiv:1603.01312*, 2016.

[9] W. Li, S. Azimi, A. Leonardis, and M. Fritz. To fall or not to fall: A visual approach to physical stability prediction. *arXiv preprint arXiv:1604.00066*, 2016.

[10] R. Mottaghi, H. Bagherinezhad, M. Rastegari, and A. Farhadi. Newtonian image understanding: Unfolding the dynamics of objects in static images. *arXiv preprint arXiv:1511.04048*, 2015.

[11] R. Mottaghi, M. Rastegari, A. Gupta, and A. Farhadi. " what happens if..." learning to predict the effect of forces in images. *arXiv preprint arXiv:1603.05600*, 2016.

[12] A. Solar-Lezama. *Program synthesis by sketching*. ProQuest, 2008.

[13] I. Sutskever, G. E. Hinton, and G. W. Taylor. The recurrent temporal restricted boltzmann machine. In *Advances in Neural Information Processing Systems*, pages 1601–1608, 2009.

[14] J. B. Tenenbaum, C. Kemp, T. L. Griffiths, and N. D. Goodman. How to grow a mind: Statistics, structure, and abstraction. *science*, 331(6022):1279–1285, 2011.

[15] T. Tieleman and G. Hinton. Lecture 6.5 - rmsprop, coursera: Neural networks for machine learning. 2012.

[16] T. Ullman, A. Stuhlmüller, and N. Goodman. Learning physics from dynamical scenes. 2014.

[17] 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 *Advances in Neural Information Processing Systems*, pages 127–135, 2015.