# Reasoning About Physical Interactions with Object-Centric Models

**Michael Janner**[1], **Sergey Levine**[1], **William T. Freeman**[2],
**Joshua B. Tenenbaum**[2], **Chelsea Finn**[1], **Jiajun Wu**[2]
[1]University of California, Berkeley [2]Massachusetts Institute of Technology
{janner,svlevine,cbfinn}@eecs.berkeley.edu {billf,jbt,jiajunwu}@mit.edu

## Abstract

Object-based factorizations provide a useful level of abstraction for interacting with the world. Building explicit object representations, however, often requires supervisory signals that are difficult to obtain in practice. We present a paradigm for learning object-centric representations for physical scene understanding without direct supervision of object properties. Our model, Object-Oriented Prediction and Planning (O2P2), jointly learns a perception function to map from image observations to object representations, a pairwise physics interaction function to predict the time evolution of a collection of objects, and a rendering function to map objects back to pixels. For evaluation, we consider not only the accuracy of the physical predictions of the model, but also its utility for downstream tasks that require an actionable representation of intuitive physics. After training our model on an image prediction task, we can use its learned representations to build block towers more complicated than those observed during training.

## 1 Introduction

Consider the castle made out of toy blocks in Figure 1a. Can you imagine how each block was placed, one-by-one, to build this structure? Humans possess a natural physical intuition that aids in the performance of everyday tasks. This physical intuition can be acquired, and refined, through experience. Despite being a core focus of the earliest days of artificial intelligence and computer vision research, a similar level of physical scene understanding remains elusive for machines.

Cognitive scientists argue that humans' ability to interpret the physical world derives from a richly structured apparatus. In particular, the perceptual grouping of the world into objects and their relations constitutes *core knowledge* in cognition (Spelke & Kinzler, 2007). While it is appealing to apply such an insight to contemporary machine learning methods, it is not straightforward to do so. A fundamental challenge is the design of an interface between the raw, often high-dimensional observation space and a structured, object-factorized representation. Existing works that have investigated the benefit of using objects have either assumed that an interface to an idealized object space already exists or that supervision is available to learn a mapping between raw inputs and relevant object properties (for instance, category, position, and orientation).

In this paper, we propose Object-Oriented Prediction and Planning (O2P2), in which we train an object representation suitable for physical interactions without supervision of object attributes. Instead of direct supervision, we demonstrate that segments or proposal regions in video frames, without correspondence between frames, are sufficient supervision to allow a model to reason effectively about intuitive physics. We jointly train a perception module, an object-factorized physics engine, and a neural renderer on a physics prediction task with pixel generation objective. We evaluate our learned model not only on the quality of its predictions, but also on its ability to use the learned representations for tasks that demand a sophisticated physical understanding.

Preprint. Work in progress.

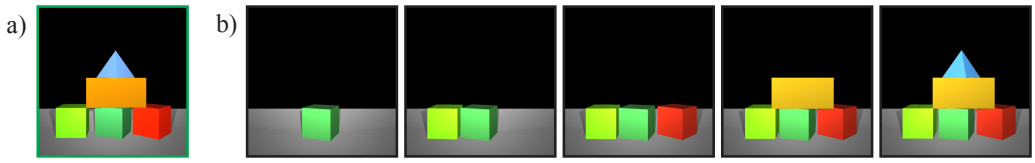


Figure 1: **(a)** A toy block castle. **(b)** Our model's build of the observed castle, using its learned object representations as a guide during planning.

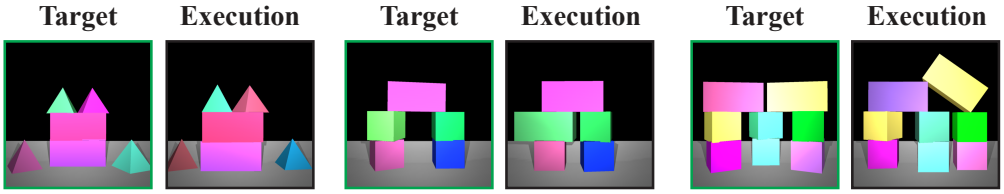


Figure 2: Given an observed block tower, O2P2 can use its learned object representations to guide planning to recreate the configuration.

## 2 Object-Oriented Prediction and Planning (O2P2)

In this section, we describe a method for learning object-based representations suitable for planning in physical reasoning tasks. O2P2 consists of three components, which are trained jointly:

- A **perception** module that maps from an image to an object encoding. The perception module is a convolutional encoder applied to each object segment independently.
- A **physics** engine to predict the time evolution of a set of objects after physics simulation. We decompose the engine into a pairwise object interaction function $f_{\text{interact}}$ and a single-object transition function $f_{\text{trans}}$, both instantiated as multi-layer perceptrons. Given $\mathbf{o}_t$, a set of object vectors from the perception module at time $t$, the physics engine outputs $\mathbf{o}_{t+1}$. For the $i^{\text{th}}$ object,
$$o_{t+1,i} = f_{\text{trans}}(o_{t,i}) + \sum_{j \neq i} f_{\text{interact}}(o_{t,i}, o_{t,j}) + o_{t,i}$$
- A **rendering** engine that produces an image prediction from a variable number of objects. We first predict a three-channel image and single-channel heatmap for each object $o_{t,i}$. We then combine all of the object images according to the weights in their heatmaps at every pixel location to produce a single composite image.

### 2.1 Planning with Learned Models

To accomplish the task depicted in Figure 1, a model must output a sequence of actions to construct an observed configuration. This setting is more challenging than simply predicting an image with a learned renderer because there is an implicit sequential ordering to building such a tower. For example, the bottom cubes must be placed before the topmost triangle. O2P2 was trained solely on a pixel-prediction task, in which it was never shown such valid action orderings (or any actions at all). However, these orderings are essentially constraints on the physical stability of intermediate towers, and should be derivable from a model with sufficient understanding of physical interactions.

The planning routine for constructing block towers is guided solely through errors in the learned object representation space. The procedure is as follows:

1. The perception module **encodes** the goal image into a set of object representations $\mathbf{o}_1^{\text{targ}}$.
2. We **sample** actions of the form `(shape, position, orientation, color)`.
3. We **evaluate** the samples by likewise encoding them as object vectors and comparing them to $\mathbf{o}_1^{\text{targ}}$. We view each action sample as an image (analogous to observing a block held in place before dropping it) and use the perception module to produce object vectors from each sample, $\mathbf{o}_0^{\text{pred}}$. Because the actions selected should produce a stable tower, we run all sampled object representations through the physics engine to yield $\mathbf{o}_1^{\text{pred}}$ before comparing to $\mathbf{o}_1^{\text{targ}}$. Object representations are compared using mean squared error (MSE).
4. Using the action sampler and evaluation metric, we **select** actions using the cross-entropy method starting from a uniform distribution. At each time step, after selecting the action that minimizes loss to one of the observed objects in the set of vectors $\mathbf{o}_1^{\text{targ}}$, we **execute** that action in MuJoCo.

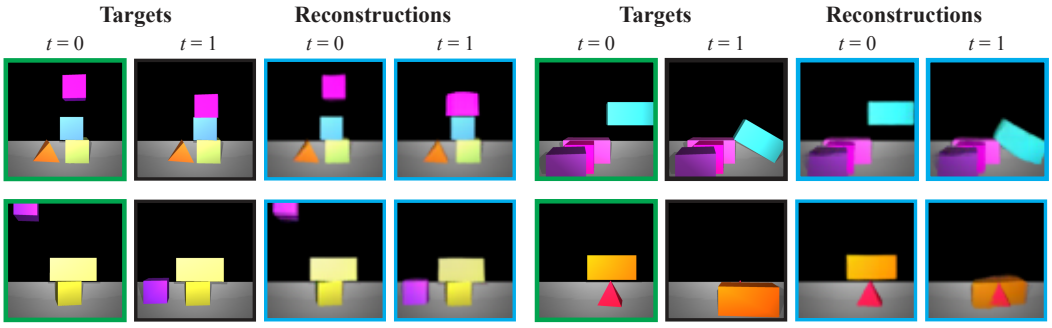


Figure 3: Given an observed image $I_0$ at $t = 0$, our model predicts a set of object representations $\mathbf{o}_0$, simulates the objects with a learned physics engine to produce $\mathbf{o}_1$, and renders the resulting predictions to get $\hat{I}_1$, the scene's appearance at a later time. Observations are outlined in green, other images rendered with the ground-truth renderer are outlined in black, and images rendered with our learned renderer are outlined in blue.

## 3 Related Work

Watters et al. (2017); Chang et al. (2016); van Steenkiste et al. (2018) have shown approaches to learning object-factorized representations from data using learned physics or interaction engines. Alternatively, other works treat physics prediction as an image-to-image translation (Lee et al., 2018) or classification (Lerer et al., 2016) problem. In contrast to these prior methods, we consider not only the accuracy of the predictions of our model, but also its utility for downstream tasks that are intentionally constructed to evaluate its ability to acquire an actionable representation of physics.

## 4 Experimental Evaluation

In our experimental evaluation, we aim to answer the following questions, (1) After training solely on physics prediction tasks, can O2P2 reason about physical interactions in an actionable and useful way? (2) Does the implicit object factorization imposed by O2P2's structure provide a benefit over an object-agnostic black-box video prediction approach? (3) Is an object factorization still useful even without supervision for object representations?

### 4.1 Learning object representations

To construct a dataset for training O2P2, we simulate dropping between one and five blocks (with randomly assigned initial position, color, and orientation) on top of a fixed platform in the MuJoCo simulator. For each training image pair $(I_0, I_1)$, we predict object representations $\mathbf{o}_0 = f_{\text{percept}}(I_0)$ from the first observation, and predict the future object representations $\mathbf{o}_1 = f_{\text{physics}}(\mathbf{o}_0)$ with the learned physics engine. The rendering engine then predicts an image $\hat{I}_t = f_{\text{render}}(\mathbf{o}_t)$ from each of the object representations. We compare each image prediction $\hat{I}_t$ to its ground-truth counterpart using both $\mathcal{L}_2$ distance and a perceptual loss in the feature space of the VGG network, $\mathcal{L}_{\text{VGG}}$ (Simonyan & Zisserman, 2014). The perception module is supervised by the reconstruction of $I_0$, the physics engine is supervised by the reconstruction of $I_1$, and the rendering engine is supervised by the reconstruction of both images. Specifically,

$$\mathcal{L}_{\text{percept}}(\cdot) = \mathcal{L}_2(\hat{I}_0, I_0) + \mathcal{L}_{\text{VGG}}(\hat{I}_0, I_0), \tag{1}$$

$$\mathcal{L}_{\text{physics}}(\cdot) = \mathcal{L}_2(\hat{I}_1, I_1) + \mathcal{L}_{\text{VGG}}(\hat{I}_1, I_1), \tag{2}$$

$$\mathcal{L}_{\text{render}}(\cdot) = \mathcal{L}_{\text{percept}}(\cdot) + \mathcal{L}_{\text{physics}}(\cdot). \tag{3}$$

Representative results on held-out random configurations are shown in Figure 3. Even when the model's predictions at $t = 1$ differ from the ground truth image (such as the bottom left example), the physics engine produces a plausible simulation of the scene from the observation at $t = 0$.

### 4.2 Matching observed towers

After training O2P2 on the random configurations of blocks, we fix its parameters and employ the planning procedure as described in Section 2.1 to build tower configurations observed in images. Qualitative results are shown in Figure 2. Our method stacks 76% of configurations correctly, as compared to 24% for a method which planned using an object-agnostic physics prediction model (Lee et al., 2018). We evaluate tower stacking success by greedily matching the built configuration to the ground-truth state of the target tower, and comparing the maximum object error (defined on its position, identity, and color) to a predetermined threshold.

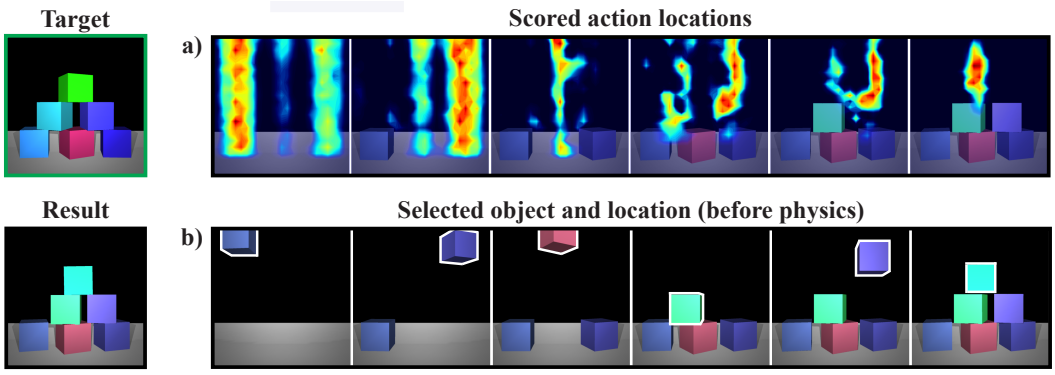


Figure 4: (a) Visualization of scored locations for dropping an object at each timestep. Because O2P2 simulates physics before selecting an action, it is able to plan a sequence of stable actions. (b) The selected block and drop position from the scored samples, outlined in white.

### 4.3 The Importance of Understanding Physics

Figure 4 depicts the entire planning and execution procedure for O2P2 on a pyramid of six blocks. At each step, we visualize the process by which our model selects an action by showing a heatmap of scores (negative MSE) for each action sample according to the sample's $(x, y)$ position (Figure 4a). Although the model is never trained to produce valid action decisions, the planning procedure selects a physically stable sequence of actions. For example, at the first timestep, the model scores three $x$-locations highly, corresponding to the three blocks at the bottom of the pyramid. It correctly determines that the height at which it releases a block at any of these locations does not particularly matter, since the block will drop to the correct height after running the physics engine. Figure 4b shows the selected action at each step.

## 5 Conclusion

We introduced a method of learning object-centric representations suitable for physical interactions. These representations did not assume the usual supervision of object properties in the form of position, orientation, velocity, or shape labels. Instead, we relied only on segment proposals and a factorized structure in a learned physics engine to guide the training of such representations. We demonstrated that this approach is appropriate for a standard physics prediction task. More importantly, we showed that this method gives rise to object representations that can be used for difficult planning problems, in which object configurations differ from those seen during training, without further adaptation. We evaluated our model on a block tower matching task and found that it outperformed object-agnostic approaches that made comparisons in pixel-space directly.

## References

Michael B Chang, Tomer Ullman, Antonio Torralba, and Joshua B Tenenbaum. A compositional object-based approach to learning physical dynamics. In *ICLR*, 2016.

Alex X. Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

Adam Lerer, Sam Gross, and Rob Fergus. Learning physical intuition of block towers by example. In *ICML*, 2016.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

Elizabeth S. Spelke and Katherine D. Kinzler. Core knowledge. *Developmental Science*, 10(1):89–96, 2007.

Sjoerd van Steenkiste, Michael Chang, Klaus Greff, and Jürgen Schmidhuber. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. In *ICLR*, 2018.

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