Interpretable Intuitive Physics Model

Tian Ye

CMU-RI-TR-19-16

May, 2019

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Abhinav Gupta, *chair*
Martial Hebert
Xiaolong Wang

*Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Robotics.*

To my friends and mentors.

Abstract

Humans have a remarkable ability to use physical commonsense and predict the effect of collisions. But do they understand the underlying factors? Can they predict if the underlying factors have changed? Interestingly, in most cases humans can predict the effects of similar collisions with different conditions such as changes in mass, friction, etc. It is postulated this is primarily because we learn to model physics with meaningful latent variables. This does not imply we can estimate the precise values of these meaningful variables (estimate exact values of mass or friction). Inspired by this observation, we propose an interpretable intuitive physics model where specific dimensions in the bottleneck layers correspond to different physical properties. In order to demonstrate that our system models these underlying physical properties, we train our model on collisions of different shapes (cube, cone, cylinder, spheres etc.) and test on collisions of unseen combinations of shapes. Furthermore, we demonstrate our model generalizes well even when similar scenes are simulated with different underlying properties.

Acknowledgments

I would first like to thank my advisor Professor Abhinav Gupta. He taught me what is research, guided me through the difficulties, and enlightened me with his vision.

I also own deep gratitude to Xiaolong Wang and Professor Saurabh Gupta for their insightful ideas and continuous support.

Finally, I must express my very profound gratitude to my parents, my boyfriend, and my friends for providing me with unfailing support and continuous encouragement throughout the two-year research experience. This work would not have been possible without them. Thank you.

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Introduction

Consider the collision image sequences shown in Figure 1.1. When people see these images, they not only recognize the shapes and color of objects but also predict what is going to happen. For example, in the first sequence people can predict that the cylinder is going to rotate while in the second sequence the ball will bounce with no motion on cylinder. But beyond visual prediction, we can even infer the underlying latent factors which can help us explain the difference in visual predictions. For example, a possible explanation of the behavior between the two sequences, if we knew the ball's mass didn't change, is that the first sequence's cylinder was lighter than the ball whereas in the second sequence the cylinder was heavier than the ball. Beyond this we can deduce that the cylinder in the first sequence was much lighter than the one in the second.

Humans demonstrate the profound ability to understand the underlying physics of the world [9, 10] and use it to predict the future. We use this physical commonsense for not only rich understanding but also for physical interactions. The question arises as to whether this physical commonsense is just an end-to-end model with intermediate representations being a black-box, or explicit and meaningful intermediate representations? For humans, the answer appears to be the latter. We can predict the future if some underlying conditions are changed. For example, we can predict that if we throw the ball in the second sequence with 10x initial speed then the cylinder might rotate.

In this paper, we focus on learning an intuitive model of physics [3, 13, 17].

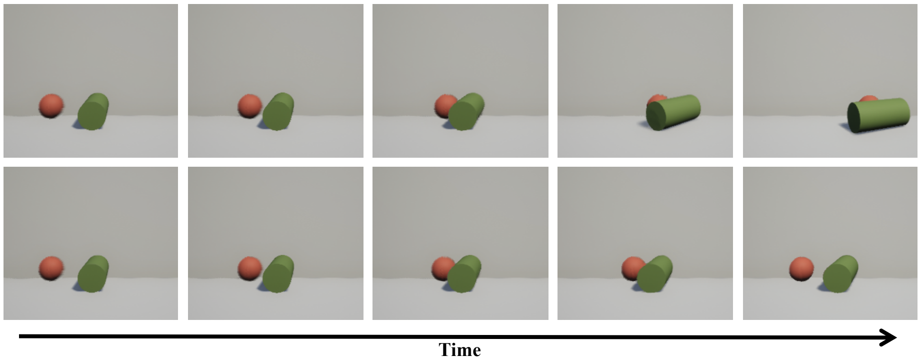

Figure 1.1: Interpretable Physics Models. Consider the sequences shown above. Not only we can predict the future frames of collisions but we can also predict the underlying factors that lead to such an inference. For example, we can infer the mass of cylinder is much higher in second sequence and therefore it hardly moves in the image. Our ability to infer meaningful underlying latent factors inspires us in this paper to learn an interpretable intuitive physics model.

Unlike some recent efforts, where the goal is to learn physics in an end-to-end manner with little-to-no constraints on intermediary layers, we focus on learning an **interpretable** model. More specifically, the bottleneck layers in our network model physical properties such as mass, friction, etc.

Learning an interpretable intuitive physics model is, however, quite a challenging task. For example, Wu et al. [26] attempts to build a model but the inverse graphics engine infers physical properties such as mass and friction. These properties are then used with neural physics engine or simulators for prediction. But can we really infer physical properties from the few frames of such collisions? Can we separate friction from mass, restitution by observing the frames? The fact is most of these physical factors are so dependent that it is infeasible to infer the exact values of physical properties. For example we can determine ratios between properties but not the precise values of both (e.g., we can determine the relative mass between two objects but not the exact values for both). This is precisely why in [26] only one factor is inferred from motion and the other factor is directly correlated to the appearance. Furthermore, the learned physics model is domain-specific and will not generalize–even across different shapes.

To tackle these challenges, we propose an interpretable intuitive physics model, where specific dimensions in the bottleneck layers correspond to different physical properties. The bottleneck layer models the distribution rather than infer precise values of mass, speed and friction. In order to demonstrate that our system models these underlying physical properties, we train our model on collision of different shapes (cube, cone, cylinder, spheres etc.) and test on collisions of unseen combinations of shapes altogether. We also demonstrate the richness of our model by predicting the future states under different physical conditions (*e.g.*, how the future frames will look if the friction is doubled).

Our contributions include: (a) an intuitive physics model that disentangles different physical properties in an interpretable way; (b) a staggered training algorithm designed to distinguish the subtleties between different physical quantities; (c) generalization to different shapes and physical quantity combinations; most importantly, (d) the ability to adapt future predictions when physical environments change. Note (d) is different from generalization: the hallucination/prediction is done for a physical scene completely different from the observed first four frames.

Chapter 2

Background

Physical reasoning and learning physical commonsense has raised a lot of interest in recent years [1, 5, 16, 17, 18, 28, 29, 31]. There has been multiple efforts to learn implicit and explicit models of physics commonsense. The underlying goal of most of these systems is to use physics to predict what is going to happen next [6, 7, 8, 13, 14, 24, 25]. The hope is that if the model can predict what is going to happen next after interacting with objects, it will be forced to understand the physical properties of the objects. For example, [13] tries to learn the physical properties by predicting whether a tower of blocks will fall. [7] proposed to learn a visual predictive model for playing billiards.

However, the first issue is what is the right data to learn this physics model. Researchers have tried a wide spectrum of approaches. For example, many researchers have focused on the task of visual prediction using real-world videos, based on the hypothesis that the predictive model will contain some underlying physical properties [15, 21, 22]. While videos provide realistic data, there is little to no control on how the data is collected and therefore the implicit models end up learning dynamic models of texture. In order to force physical commonsense learning, people have even tried using videos of physical interactions. For example, Physics101 dataset [25] collects sequences of collisions for this task. But most of the learning still happens passively (random batches). In order to overcome that, recent approaches have tried to learn physics by active interaction using robots [1, 6, 18]. While there is more control in the process of data collection, there are still issues with lack of diverse data due to most experiments being performed in lab setting with few objects. Finally, one can collect data with full control over several physical parameters using simulation. There has been lot of recent efforts in using simulation to learn physical models [7, 13, 16, 17]. One limitation of these approaches, in terms of data, is the lack of diversity during training, which forces them to learn physics models specific to particular shapes such as blocks, spheres etc. Furthermore, none of these approaches use the full power of simulation to generate a dense set of videos with multiple conditions. Most importantly, none of these approaches learn an interpretable model.

Apart from the question of data, another core issue is how explicit is the representation of physics in these models. To truly understand the object physical properties, it requires our model to be interpretable [2, 4, 12, 23, 26]. That is, the model should not only be able to predict the futures, but the latent representations should also indicate the physical properties (e.g., mass, friction and speed) implicitly or explicitly. For example, [2] proposed an Interaction Network which learns to predict the rigid body dynamics of gravitational systems. [26] proposed to explicitly estimate the physical object states and forward this state information to a physics engine for prediction. However, we argue exact values of these physical properties might not be possible due to entanglement of various factors. Instead of estimating the physics states explicitly, our work focuses on separating the dimensions in the bottleneck layer.

Our work is mostly related to the Inverse Graphics Network [12]. It learns a disentangled representation in the graphics code layer where different neurons are encouraged to represent different transformations including pose and light. The system can be trained in an end-to-end manner without providing an explicit state value as supervisions for the graphics code layer. However, unlike the Inverse Graphics Network, where pose and light can be separately inferred from the input images, the dynamics are dependent on the joint set of physical properties in our model (mass, friction and speed), which confound future predictions.

Our model is also related to the visual prediction models [11, 15, 19, 20, 22, 27, 30] in computer vision. For example, [20] proposed to directly predict a sequence of video frames in raw pixels given a sequence of former frames as inputs. Instead of directly predicting the pixels, [22] proposed to predict the optical flows given an input image and then warp the flows on the input images to generate future frames. However, the optical flow estimation is not always correct, introducing errors in the supervisions for training. To tackle this, [30] proposed a bilinear sampling layer which makes the warping process differentiable. This enables them to train their prediction model from pixels to pixels in an end-to-end manner.

Chapter 3

Dataset

We create a new dataset for our experiments in this paper. The advantage of our proposed dataset is that we have rich combinations of different physical properties as well as different object appearances for different types of collisions (falling over, twisting, bouncing, etc.). Unlike previous datasets, the physical properties in our dataset are independent from the object shapes and appearance. In this way, we can train models which force estimation of physical properties by observing the collisions. More importantly, our testing sets contain novel combinations of object shapes or physical properties that are unseen in the training set. The details of dataset generation is illustrated as following.

We generate our data using the Unreal Engine 4 (UE4) game engine. We use 11 different object combinations with 5 unique basic objects: sphere, cube, cylinder, cone, and wedge. We select 3 different physical properties including mass of static object, initial speed of colliding object and friction of floor. For each property, we choose 5 different scales of values as shown in Table 3.1. For simplicity, we specify a certain scale of parameter by the format $\{parameter\ name\}_{\{scale\}}$ (e.g., $mass_1$, $friction_4$, $speed_2$). We simulate all the $5 \times 5 \times 5 = 125$ sets of physical combinations. For each set of physical property combination, there are 11 different object combinations and 15 different initial rotation and restitution. Thus in total there are $125 \times 15 \times 11 = 20625$ collisions. Each collision is represented by 5 sample frames with 0.5s time intervals between them.

The diversity in our dataset is highlighted in Figure 3.1. For example, our dataset

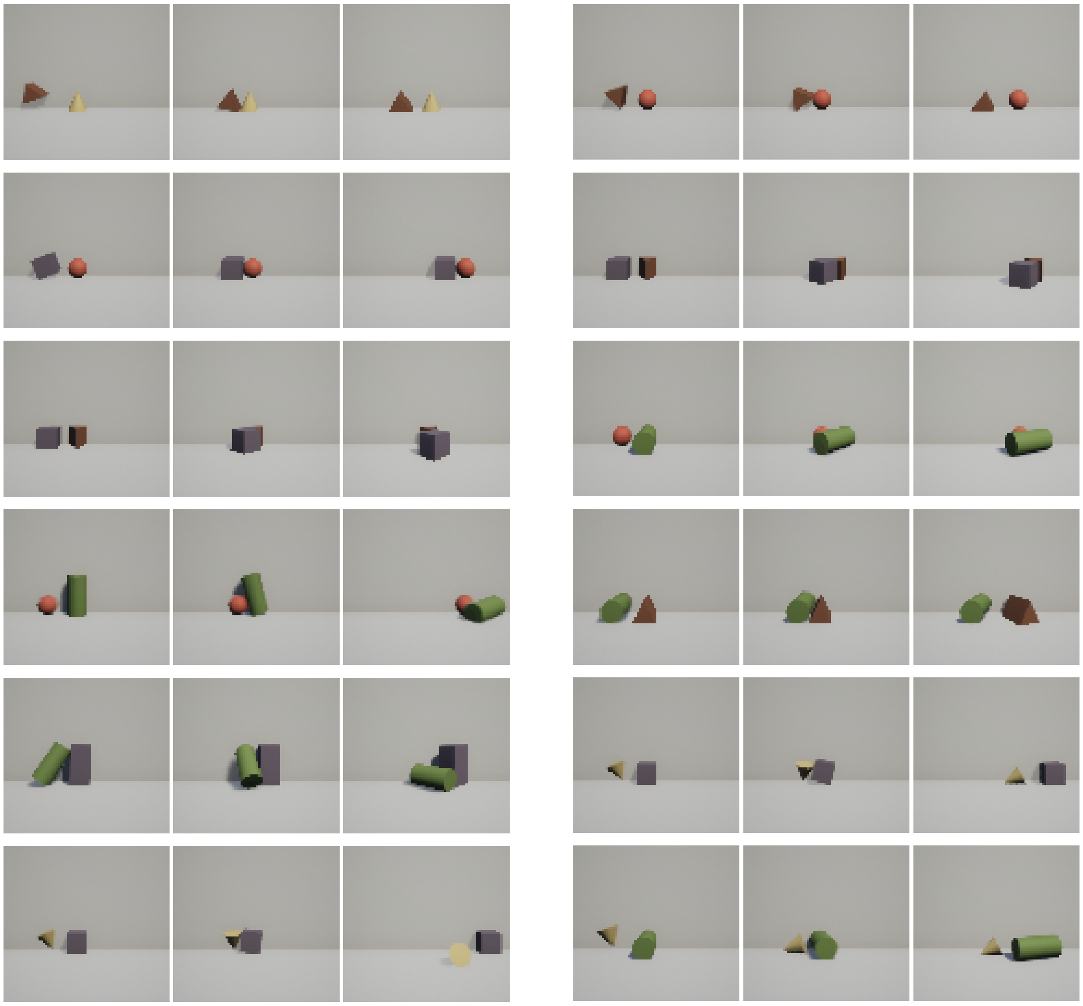

Figure 3.1: Our dataset includes 2 object collisions with a variety of shapes. Unlike existing physics datasets which have only one type of shape, our dataset is diverse in terms of different shapes and physical properties of objects.

has cones toppling over; cylinders falling down when hit by a ball and rolling cylinders. We believe this large diversity makes it one of the most challenging datasets to learn and disentangle physical properties.

For training, we use 124 sets of physics combination with 9 different object combinations (16740 collisions). The remaining data are used for two types of testing: (i) parameter testing and (ii) shape testing. The parameter testing set contains 135 collisions with unseen physical parameter combinations ($mass_3$, $speed_3$, $friction_3$) but seen object shape combinations. The shape testing set on the other hand,

Table 3.1: Dataset Settings

	$scale_{\mathbf{1}}$	$scale_{\mathbf{2}}$	$scale_{\mathbf{3}}$	$scale_{\mathbf{4}}$	$scale_{\mathbf{5}}$
Mass	100	200	300	400	500
Speed	10000	20000	30000	40000	50000
Friction	0.01	0.02	0.03	0.04	0.05

contains 3750 collisions with 2 unseen shape combinations yet seen physical parameter combinations. We show the generalization ability of our physics model on both testing conditions.

Chapter 4

Interpretable Physics Model

Our goal is to develop a physics-based reasoning network to solve prediction tasks, *e.g.*, physical collisions, while having interepretable intermediate representations.

4.1 Visual Prediction Model

As illustrated in Figure 4.1, our model takes in 4 RGB video frames as input and learns to predict the future 5th RGB frame after the collisions. The model is composed with two parts: an encoder for extracting abstract physical representations and a decoder for future frame prediction.

4.1.1 Encoder for physics representations

The encoder is designed to capture the motion of two colliding objects, from which the physical properties can be inferred. Given 4 RGB frames as inputs, they are first forwarded to a ConvNet with AlexNet architecture and ImageNet pre-training. We extract the pool5 feature for each video frame and concatenate the features together as a representation for the input sequence. This feature is then forwarded to two convolutional layers and four fully connected layers to obtain the physics representation.

The physics representation is a 306 dimensional vector, which contains disentangled neurons of mass (dimensions 1 to 25), speed (dimensions 26 to 50), friction (dimensions

51 to 75), and other intrinsic information (dimensions 76 to 306), as shown in Figure 4.1. Note that although the vector is disentangled, there is no explicit meanings for each neuron value.

4.1.2 Decoder for future prediction

The physics representation is forwarded to a decoder for future frame prediction. Our decoder contains one fully-connected layer followed by six decovolutional layers. Inspired by [22, 30], our decoder uses optical flow fields as the output representation instead of directly outputing the RGB raw pixel values. The optical flow is then used to perform warping on the last input frame by a bilinear sampling layer [30] to generate the future frame. Since the bilinear sampling layer is differentiable, the network can be trained in an end-to-end manner with the 5th frame for direct supervision.

There are two major advantages of using optical flow as outputs: (i) it can force the model to learn the factors that cause the changes between two frames; (ii) it allows the model to focus on the changes of the foreground objects.

4.2 Learning Objective

Formally, we define the encoder as a function f and the decoder as a function g. Given an image sequence x as inputs (4 frames), our encoder transforms the images into a physically meaningful and disentangled representation $z = f(x)$ and then the decoder transforms this representation into a future frame $y = g(z)$.

The disentangled representation z can be formulated as $z = (\phi^m, \phi^s, \phi^f, \phi^i)$ where $(\cdot, \cdot)$ denotes concatenation. The first part (ϕ^m, ϕ^s, ϕ^f) denotes the combination *physics variable*, which encodes the physical quantities (m, s, f stands for mass, speed, and friction respectively). The second part ϕ^i is the *intrinsic variable*, representing all the other intrinsic properties in the scene (*e.g.*, colors, shapes and initial rotation).

In this paper, we study the effect of varying the values of physical quantities in a two-object collision scenario. Following the strategy in [12], we group our training sequence samples into mini-batches. Inside one mini-batch, only one physical property changes across all the samples and other physical properties remain fixed. We denote

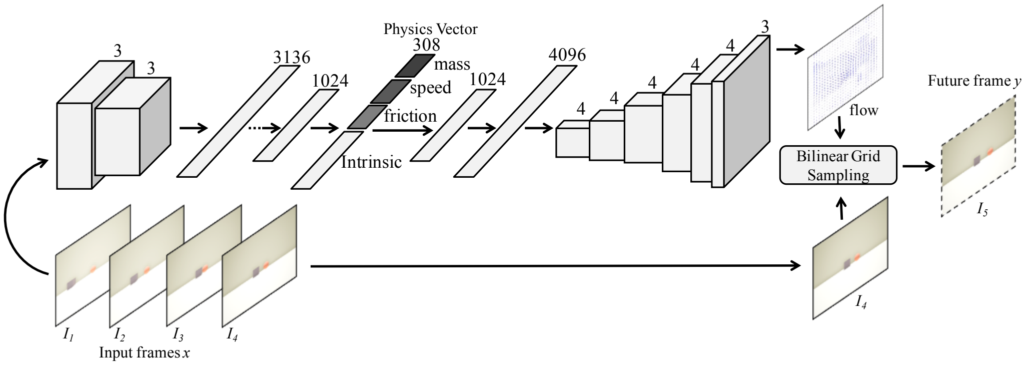

Figure 4.1: Model Architecture: we follow an encoder-decoder framework. The encoder takes 4 frames of a collision (2 before collision, 1 during collision, and 1 after collision). All inputs are first passed through a pre-trained Alexnet. The Alexnet features are further appended along channels and are sent to two convolution layers and four fully-connected layers. The resulting physics vector is passed through a decoder consisting of one fully-connected layer and six up-sampling convolution layers to produce an optical flow. The number on the convolution layers and transpose convolution layers stands for the kernel size of corresponding layer. The last bilinear grid sampling layer takes the optical flow and the 4^{th} input frame to produce future prediction.

$B^p = \{(x_k, y_k)\}_{k=1}^{5}$ as one mini-batch with 5 sequences, where the only changing property is p (i.e., we use p as a variable to represent either mass, speed or friction).

For each mini-batch B^p during training, we encourage only the dimensions corresponding to the property p to change in z. For example, when training with a mini-batch where only mass is changing, we force the network to have different values in the dimensions for ϕ^m and same values for the rest of the dimensions in z. For simplicity, we further denote the dimensions which relevant to p in z as ϕ_k^p and the rest of the dimensions as $\bar{\phi}_k^p$ for example k.

We train our prediction model with this constraint. Assuming we are training with one batch $B^p = \{(x_k, y_k)\}_{k=1}^{5}$. In a maximum likelihood estimation (MLE) framework, this can be formulated as maximizing the log-probabilities under the desired constraints:

$$\begin{aligned} &\text{maximize} \quad \sum_{k=1}^{5} \log(\mathrm{P}(y_k|x_k)) \\ &\text{subject to} \quad \bar{\phi}_i^p = \bar{\phi}_j^p, \forall 1 \leq i, j \leq 5 \end{aligned} \tag{4.1}$$

where $\bar{\phi}_k^p$ contains both the intrinsic variable inferred from image sequence x_k and inferred physics variables, except for the changing parameter.

In our auto-encoder architecture, the objective function is equivalent to minimizing the l1 distance between the predicted images $\hat{y}_k$ and the ground truth future images y_k:

$$\mathcal{L}_{mle} = \sum_k ||\hat{y}_k - y_k||_1. \tag{4.2}$$

The constraints in Eq. 4.1 can be satisfied via minimizing the loss between $\bar{\phi}_k^p$ and the mean of them within the mini-batch $\bar{\phi}^p = \frac{1}{5} \sum_k \bar{\phi}_k^p$ as,

$$\mathcal{L}_{ave} = \sum_k ||\bar{\phi}_k^p - \bar{\phi}^p||_2^2. \tag{4.3}$$

We apply both losses jointly during training our model with a constant λ balancing between them as,

$$\mathcal{L} = L_{mle} + \lambda L_{ave}. \tag{4.4}$$

In practice, we set the λ dynamically so that both gradients are maintained in the same magnitude. The value of λ is around $1e-6$.

4.3 Staggered Training

Although we follow the training objective proposed in [12], it is actually non-trivial to directly optimize with this objective. There is a fundamental difference between our problem and the settings in [12]: the physical dynamics are dependent across the set of properties, which confounds training. The same sequence of inputs and output ground-truth might infer different combinations of the physical properties. For example, both large friction and slow speed can lead to small movements of the second object after collision. Thus modifications on training method is required to handle this multi-modality issue.

We propose a staggered training algorithm to alleviate this problem. We first divide the entire training set D into 3 different sets $\{D^p\}$, where p indicates one of the physics properties(mass, speed or friction). Each D^p contains different mini-batches of B^p, inside which the only changing property is indicated by p.

The idea is: instead of training with all the physics properties at the same time in the beginning, we perform curriculum learning. We first train the network with one subset D^p and then progressively add more subsets with different properties into training. In this way, our training set becomes larger and larger through time. By learning the physics properties in this sequential manner, we force the network to recognize new physical properties one by one while keeping the learned properties. In practice, we observe that in the first training session, the network behaves normally. For the following training sessions, the loss will increase in the beginning, and will decrease to roughly the same level as the previous session.

Chapter 5

Experiments

We now demonstrate the effectiveness and generalization of our model. We will perform two sets of experiments with respect to two different testing sets in our dataset. One tests on unseen physical property combinations but seen shape combinations, and the other tests on unseen shape combinations with seen physical properties. Before going into further analysis, we will first describe the implementation details of our model and the baseline method.

5.1 Implementation details

In total, we trained for 319 epochs. We used ADAM for optimization, with initial learning rate 10^{-6}. During training, each mini-batch mentioned above has 5 sequences. During the training for the first physical quantity, each batch contains 3 mini-batches, which means 15 data in total. For the second round of staggered training, each batch contains 2 mini-batches, one for each physical quantity; similarly, in the third round of training, each batch contains 3 mini-batches, one for each physical quantity.

5.2 Baseline model

Our baseline model learns intuitive physics in an end-to-end manner and post-hoc obtains the dimensions that correspond to different physical properties. We need the disentagled representation because we want to test the generalization when the

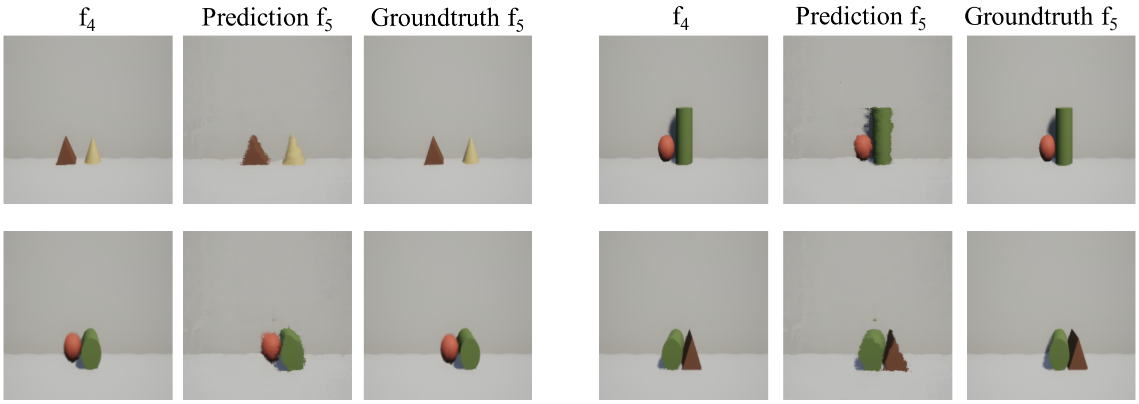

Figure 5.1: Prediction results for unseen parameters but seen shapes.

physical properties are different from input video: *e.g.*, what happens if friction is doubled? What happens if the speed is 1/10th?

For the baseline, we use the same network architecture. Different from our approach, we do not add any constraints on the bottleneck representation layer as in Eq. 4.1 in the baseline model. However, we still want to obtain the disentangled representation from this baseline for comparison. Recall that we have a subset D^p for each property p (mass, friction or speed). The examples in each mini-batch inside D^p specify the change of property p. We compute the variances for each neuron in the bottleneck representation for each D^p, and select 25 dimensions with top variances as the vector indicating property p.

5.3 Visual prediction

5.3.1 Unseen Parameters

First we evaluate if we can predict future pixels when we see a novel combination of physical parameters. Specifically, our model has never seen in training a combination of mass=3, friction=3 and speed=3. Figure 5.1 shows our interpretable model generalizes well and produces high quality predictions.

5.3.2 Unseen Shape Combinations

Next, we want to explore if our visual prediction model generalizes to different shape combinations using two unseen sets: (a) cone and cuboid; (b) cuboid and sphere. To demonstrate that our model understands each of these physical properties, we show contrasted prediction results for two different values. For example, we will use different friction values $(1, 5)$ but same mass and speed. Comparing these two outputs should highlight how our approach understands the underlying friction values.

As shown in Figure. 5.2, our predicted future frame has high quality compared to the ground-truth. We show that our model can generalize the physics reasoning to unseen objects and learn to output different collisions results given different physical environments. For example in the second condition, when the mass of sphere is high (5), our approach can predict it will not move and instead the cube will bounce back. We also compare our approach to baseline quantitatively: our approach has pixel error of 87.3, while baseline has pixel error of 95.6.The results clearly indicate our interpretable model tends to generalize better than an end-to-end model when test conditions are very different.

In addition to the baseline, we also compare our model with two other methods based on optical flow. First, we trained another prediction network using the optical flow computed between the 4th and the 5th frame as direct supervisions, instead of using the pixels of the 5th frame. For testing, we apply the predicted optical flows on the 4th frame to generate the future frame. The loss between the future frame and the ground-truth 5th frame is 118.8. Second, we computed 3 optical flows of first 4 frames, using which to find a linear model to generate the future optical flow. We apply this optical flow on the 4th frame and compare the result to the ground-truth 5th frame. The error reaches to 292.5. The result shows that our method achieves high precision than using optical flow directly.

5.4 Physical Interpolation

To show our model has actually learnt physics properties, we perform a series of interpolations on the bottleneck representation.

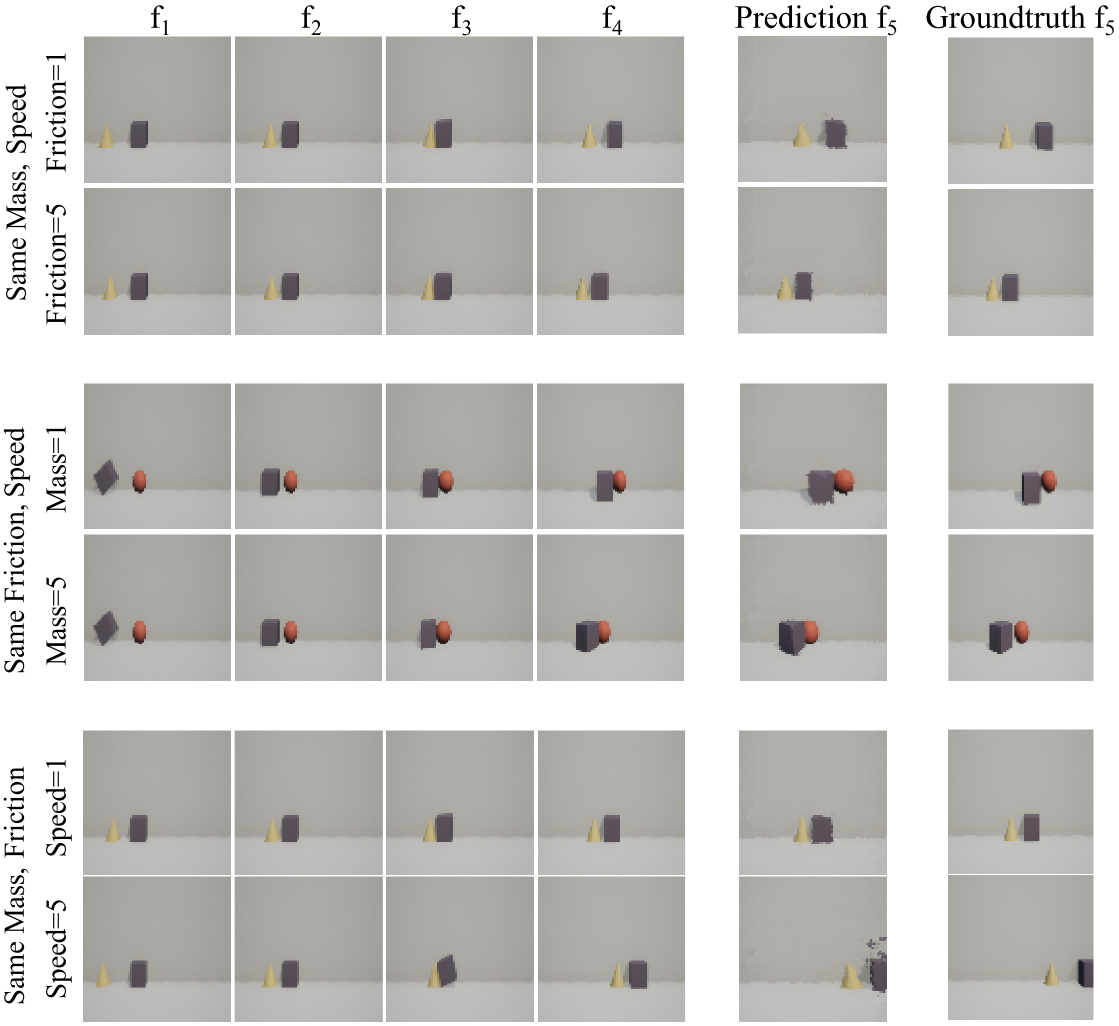

Figure 5.2: 4 input frames, the predicted 5th frame and ground-truth for collisions with unseen shape combinations. Contrast the predictions as one of physical property changes. For example, to show our approach understand these shapes, we predict for two different friction values in first case (keeping mass and speed same). The less motion in 2nd case shows that our approach understands the concept of friction.

Table 5.1: Interpolation Result. The numbers are pixel prediction errors

Method	shape 2	shape 3	shape 4	shape 5	parameter 3
Baseline	117.76	130.41	154.78	173.80	299.88
Flow + Physics	272.02	317.79	328.06	336.54	671.51
Ours	**110.93**	**119.73**	**131.70**	**138.04**	**154.09**

5.4.1 Interpolating physics representation within a mini-batch

We first show that the learned bottleneck layer is meaningful and smooth. To demonstrate this, we interpolate between different physical properties and compare our result with the ground-truth. The experiment is conducted in the following way. Let's take mass as an example: given a mini-batch where only mass changes, we use the encoder to get the physics vector $z_1 = (\phi_1^m, \phi_1^s, \phi_1^f, \phi_1^i)$ from mass$_1$ data and $z_5 = (\phi_5^m, \phi_5^s, \phi_5^f, \phi_5^i)$ from mass$_5$ data. To estimate the physics vector for mass$_i$, we interpolate a new mass variable $\hat{\phi}_i^m = (1 - 0.25i) \cdot \phi_1^m + 0.25i \cdot \phi_5^m$ and use this to create a new physics vector $\hat{z}_i = (\hat{\phi}_i^m, \phi_1^s, \phi_1^f, \phi_1^i)$. We pass the new vector to the decoder to predict the optical flows, which are warped to the 4th image in sequence i via the bilinear sampling layer, and generate the future frame.

We perform the same set of experiments for the baseline model. Quantitatively, we evaluate the prediction using the sum of mean square error for each pixel, as shown in Table 5.1, which shows that our method is significantly better than the baseline. We also visualized the results in Figure 5.3. Interestingly, our interpolation results are also very close to the ground-truth. On the other hand, baseline models failed easily when there is a dramatic change during interpolations.

We also trained another model which takes physics parameters and the optical flows of first 4-frame as inputs, and predicts the future frame. This model performs much worse than our model in the interpolation test as shown in Figure 5.3. We believe a ground-truth physics parameter based approach focuses on classification instead of learning an intuitive physics model. In interpolation experiments, the model cannot separate physics information from the optical flow features.

From these comparison, we can see that only by learning interpretable representations, we can generate reasonable prediction results after interpolations.

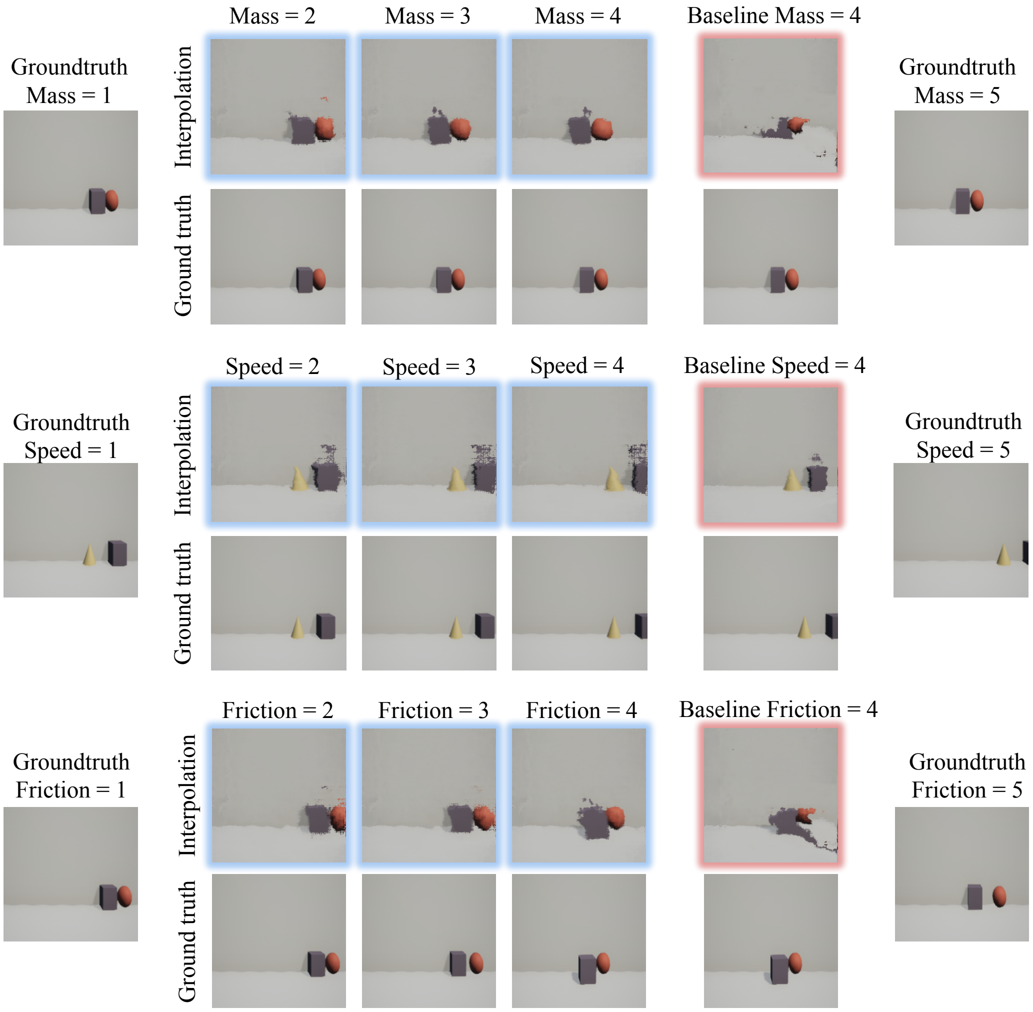

Figure 5.3: Interpolation results for physical quantity with different values. Our interpolation results are shown with blue frames. Images with red frame in last column represents the interpolation results for baseline when physical quantities equal to 4.

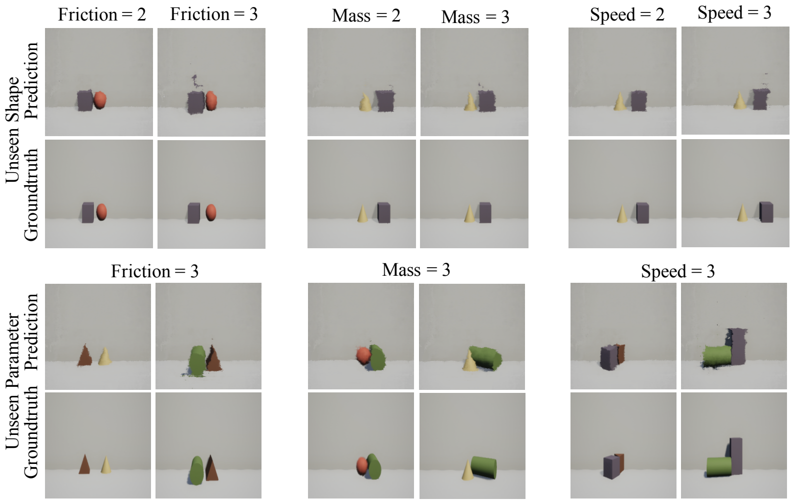

Figure 5.4: Prediction by learning double, triple ratio relation for different physical entities. Top: the result with unseen shapes. Bottom: result with unseen parameters.

5.4.2 Changing physical properties

In this experiment, we show that physics variables learned by our model are interpretable by finding a mapping between different scale of the same physical property. Specifically, we want to see: can we predict the future if the mass is doubled while all other physics conditions remain the same? For each physical quantity p, we train two networks F_2^p and F_3^p which learns to double or triple the scale of a physical property. For example, we can project the physics representation of mass$_1$ to mass$_3$ by using the network F_3^p. The network architecture for both F_2^p and F_3^p is a simple 2-layer fully connected network with 256 hidden neurons per layer. These two networks can be trained using the physical representations inferred by our encoder with the training data.

In testing time, we apply the similar interpolation as the last experiment. The only difference is that instead of using an interpolation between two relevant representations, we use the fully connected network to generate the new representations. We again evaluate the quantitative results by computing the mean square error over the pixels. As shown in Table 5.2, we have a larger performance gain in this setting compared to the baseline. Figure 5.4 shows the prediction results of our model when the physics property is enlarged from scale 1 to 2 and 3, which are all very close to

Table 5.2: Ratio Result. Comparing visual prediction when underlying physical parameters are changed by a factor

Method	shape ratio 2 (↓)	shape ratio 3 (↓)	parameter ratio 3 (↓)
Baseline	345.60	310.37	490.92
Ours	**110.79**	**124.00**	**157.10**

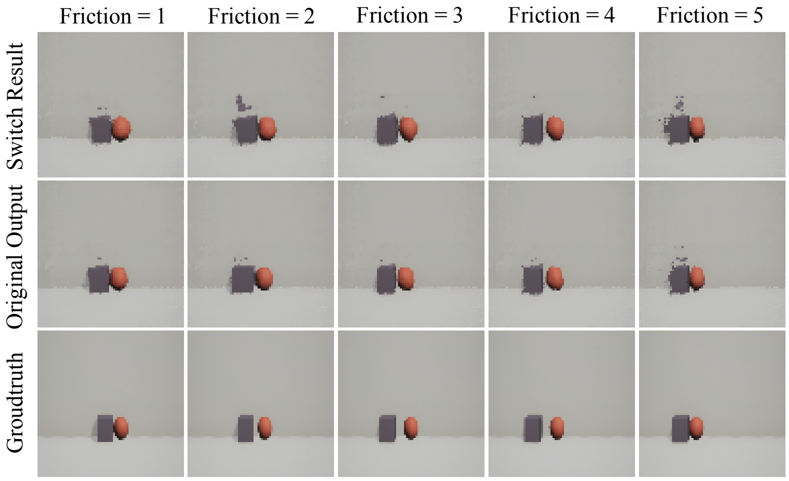

Figure 5.5: Prediction when physical property vector from one shape combination is applied to a different shape combinations. The first row shows switched result; the second row shows the prediction without switching; the third row shows ground-truth.

the ground-truth. This is another evidence showing our physics representation is interpretable and generalizes significantly better.

5.4.3 Switching between the object shapes

In experiments above, we interpolate the physics representation and apply them to the same object shape combinations. In this experiment, for a physical property p, we replace the corresponding variable ϕ^p of one collision with the variable from another collision with different objects but the same p value. We visualize the results in Figure 5.5, where the first line shows the predictions when we replace current ϕ^p with one from another shape combination. The results are almost same as the original prediction and the ground-truth, which means that the physical variable of same value can be transferred among different shape combinations. It also shows that the dimensions of physics and other dimensions are independent and can be appended easily.

Chapter 6

Conclusions

We demonstrated an interpretable intuitive physics model that generalizes across scenes with different underlying properties and object shapes. Most importantly, our model is able to predict the future when physical environment changes. To achieve this we proposed a model where specific dimensions in the bottleneck layers correspond to different physical properties. However, often physical properties are dependent and intertangled, so we introduced a training curriculum and generalized loss function that was shown to outperform the baseline approaches.

Appendix A

Appendix

A.1 Visual Prediction Results

A.1.1 friction

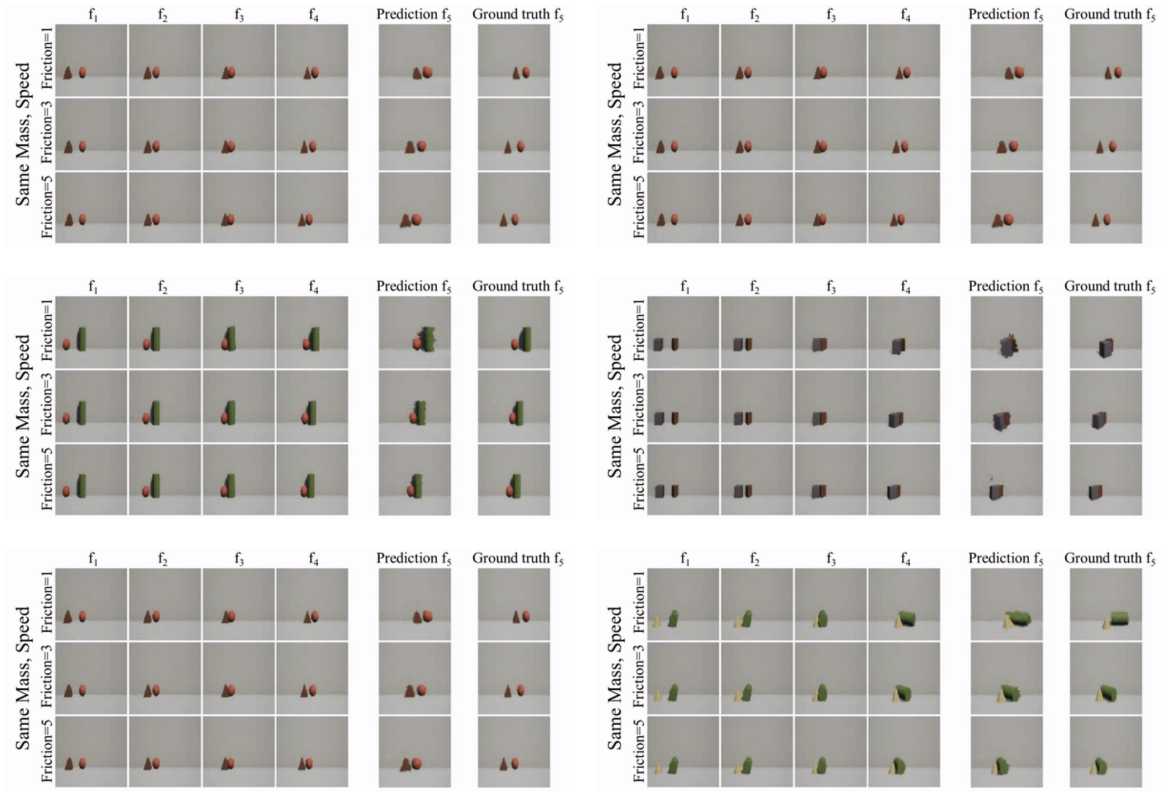

Figure A.1: Prediction results for different frictions with different objects

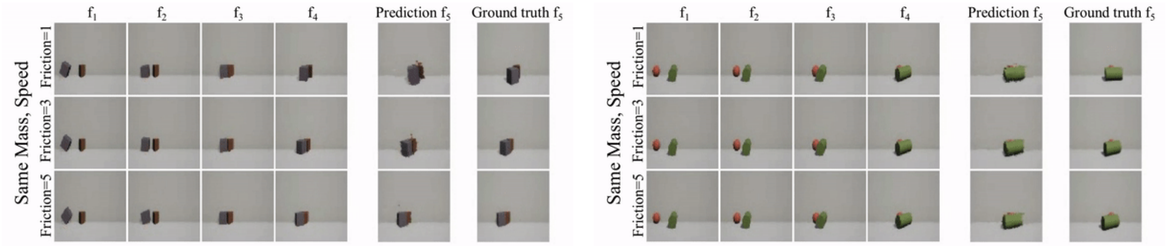

Figure A.2: Prediction results for different frictions with different objects

A.1.2 mass

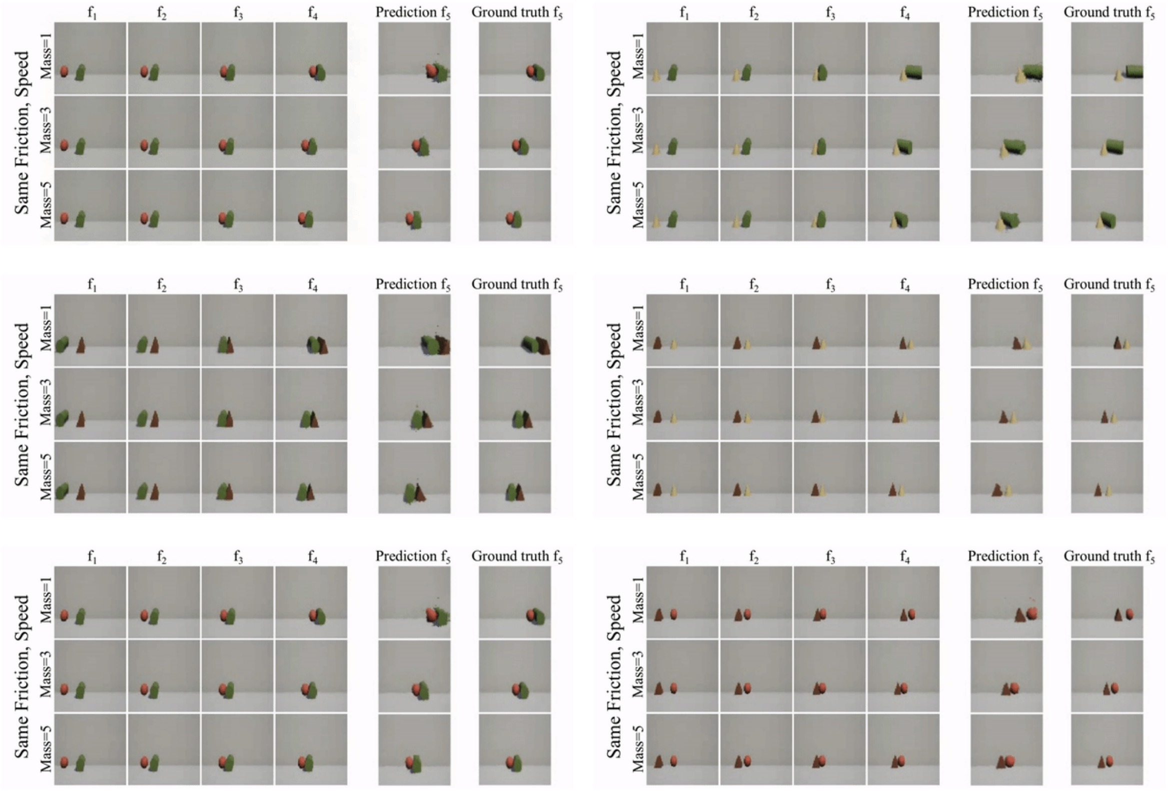

Figure A.3: Prediction results for different mass with different objects

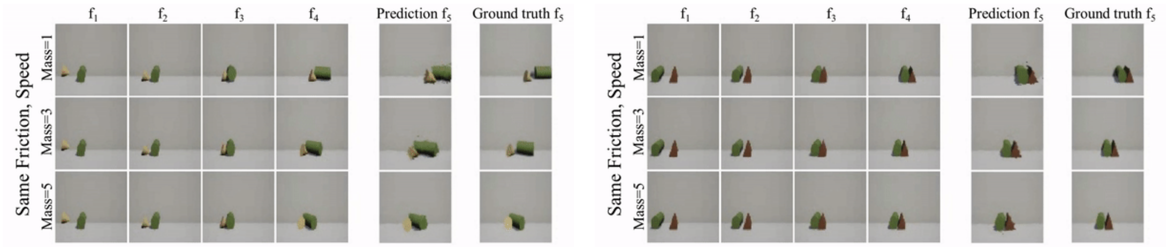

Figure A.4: Prediction results for different mass with different objects

A.1.3 speed

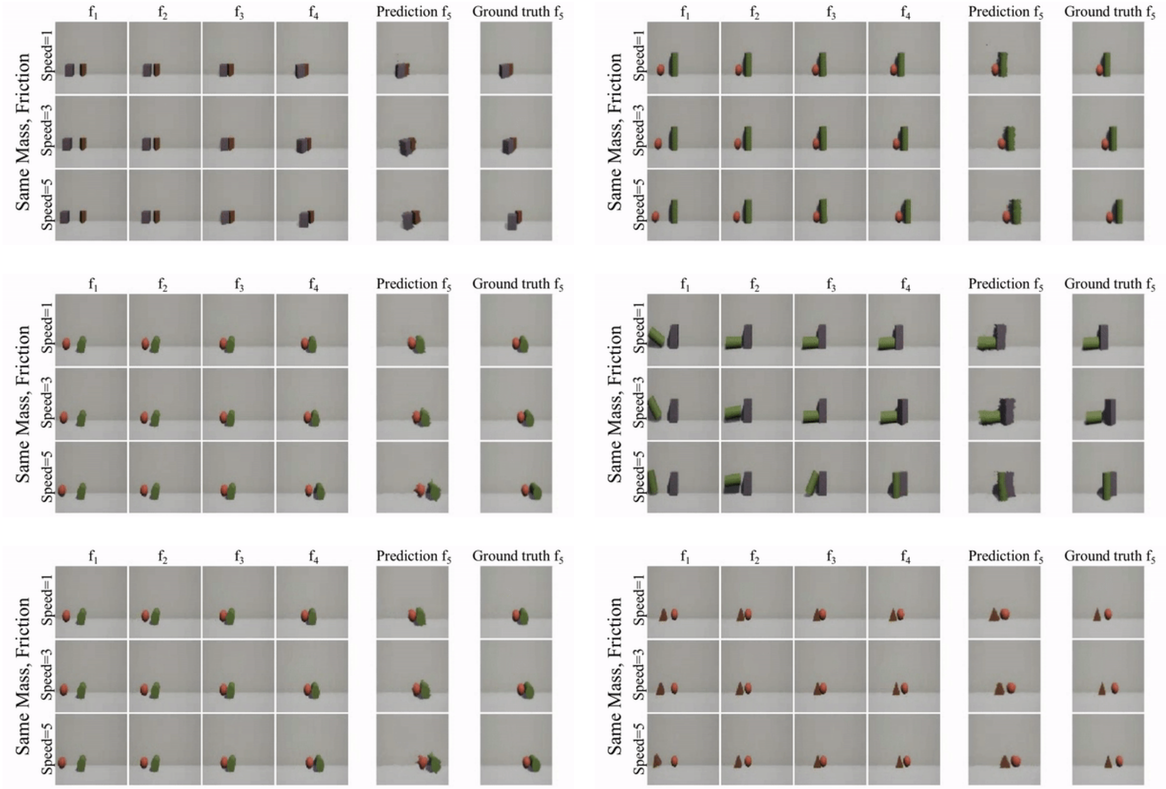

Figure A.5: Prediction results for different speeds with different objects

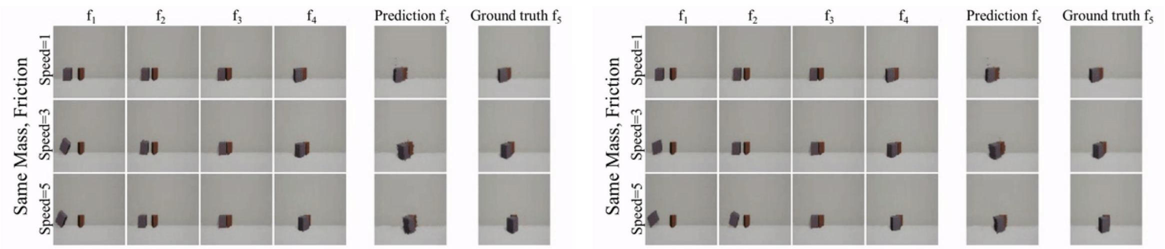

Figure A.6: Prediction results for different speeds with different objects

A.2 Physics Interpolation Results

A.2.1 friction

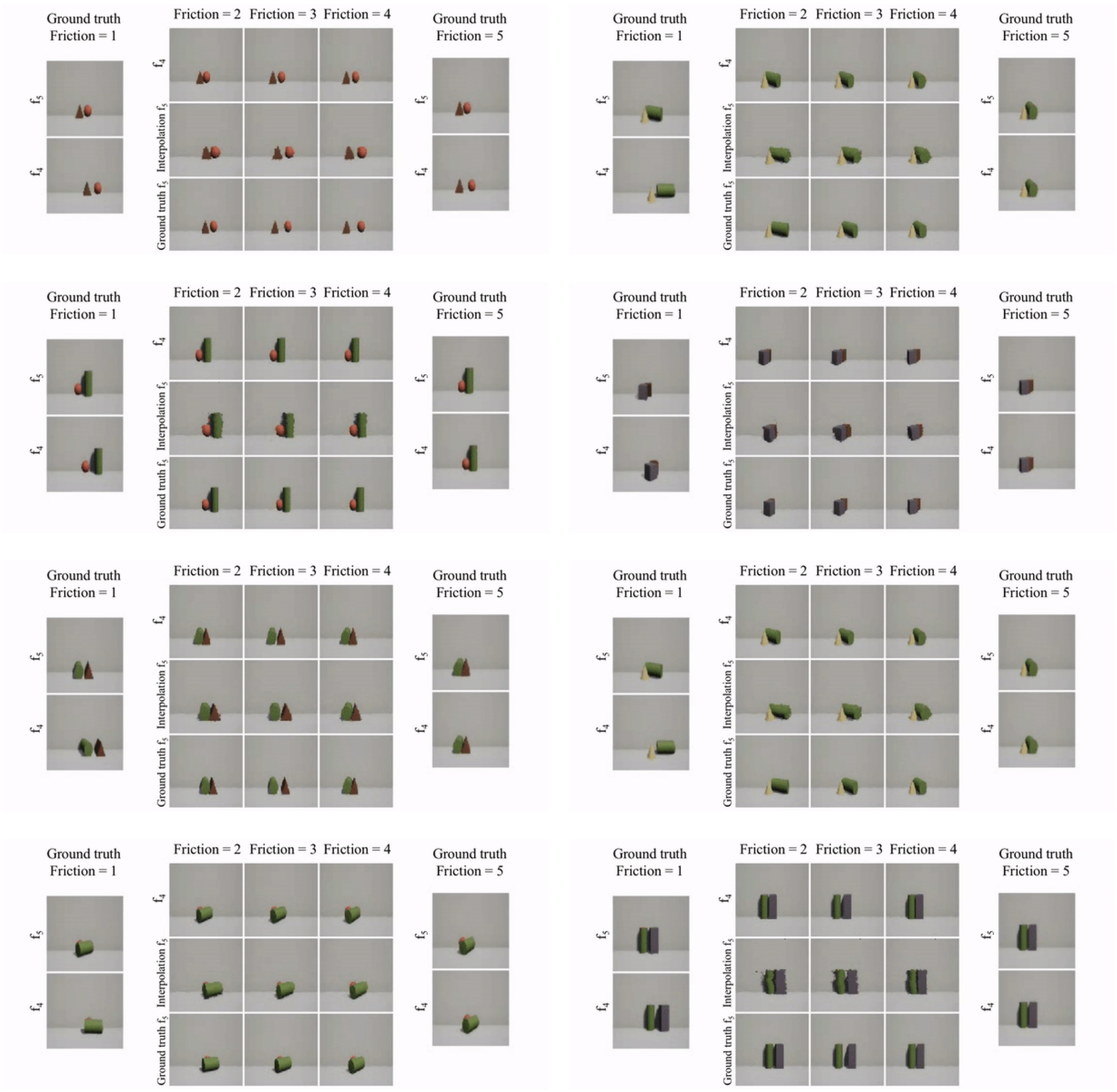

Figure A.7: Interpolation results for different friction with different objects

A.2.2 mass

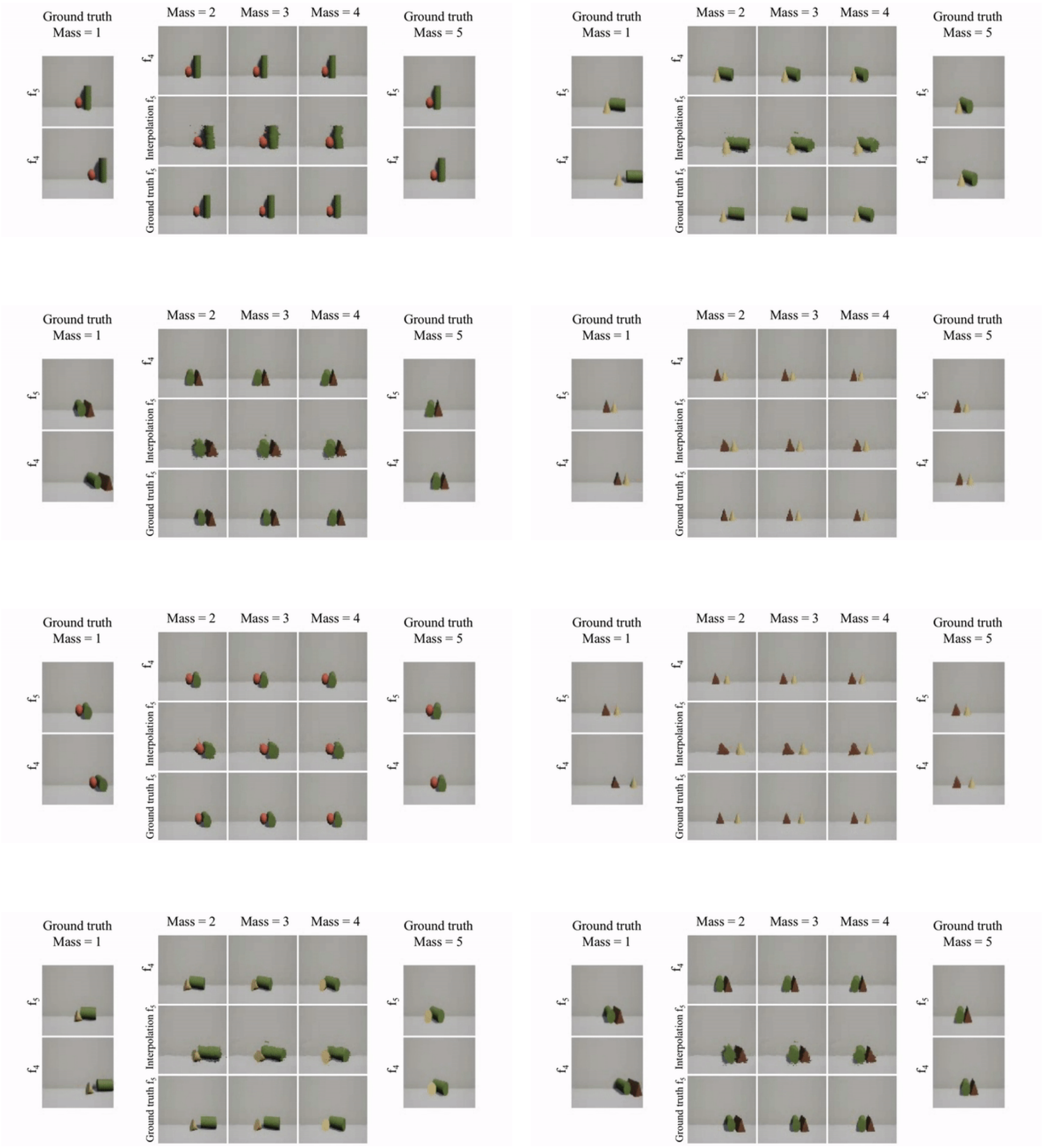

Figure A.8: Interpolation results for different friction with different objects

A.2.3 speed

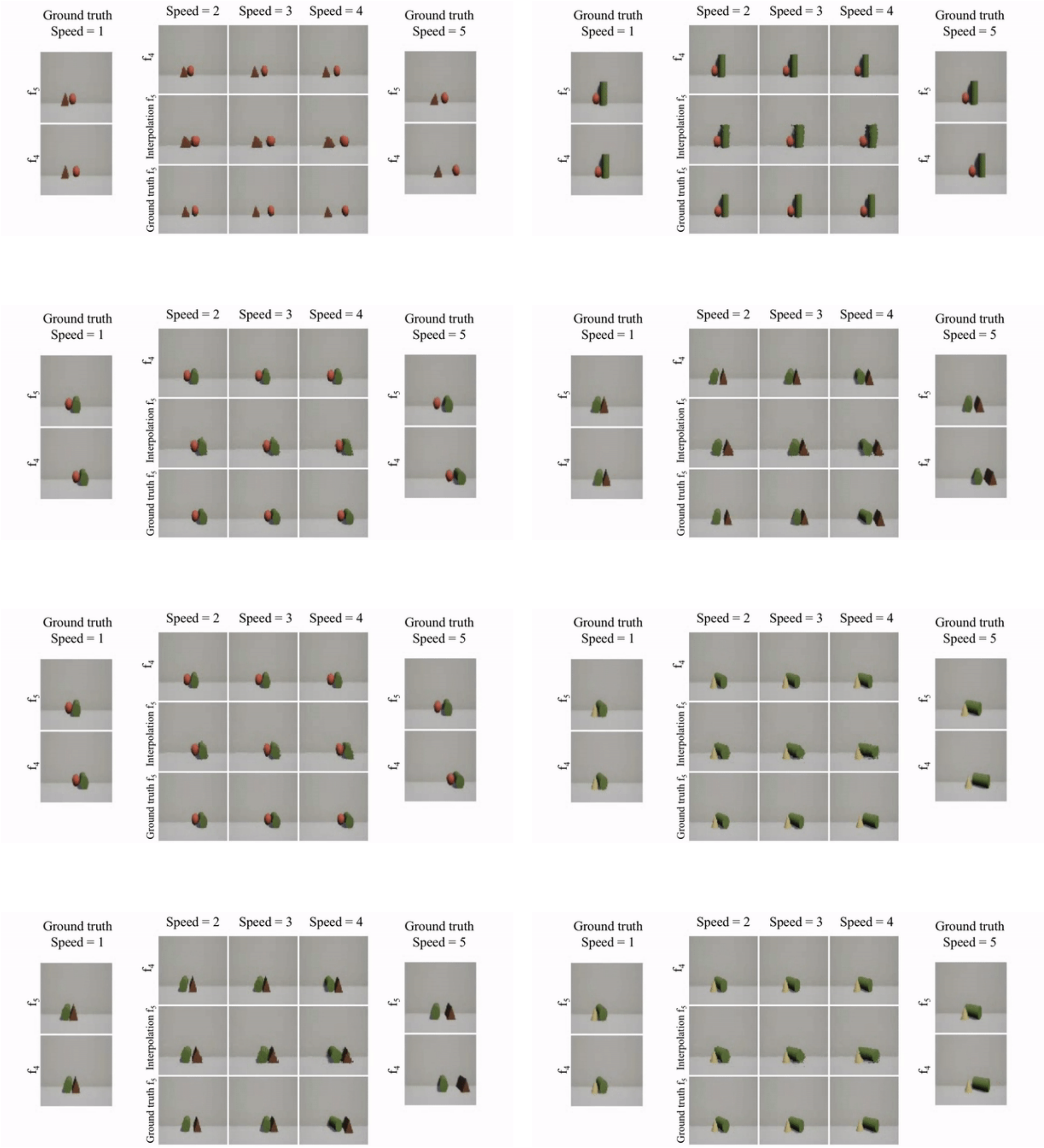

Figure A.9: Interpolation results for different speeds with different objects

Bibliography

[1] Pulkit Agrawal, Ashvin Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. In *Neural Information Processing Systems (NIPS)*, 2016. 2

[2] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. In *Neural Information Processing Systems (NIPS)*, 2016. 2

[3] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. In *Neural Information Processing Systems (NIPS)*, 2016. 1

[4] Michael B Chang, Tomer Ullman, Antonio Torralba, and Joshua B Tenenbaum. A compositional object-based approach to learning physical dynamics. In *International Conference on Learning Representations (ICLR)*, 2017. 2

[5] Mark Edmonds, Feng Gao, Xu Xie, Hangxin Liu, Siyuan Qi, Yixin Zhu, Brandon Rothrock, and Song-Chun Zhu. Feeling the force: Integrating force and pose for fluent discovery through imitation learning to open medicine bottles. In *Intelligent Robots and Systems (IROS)*, 2017. 2

[6] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Neural Information Processing Systems (NIPS)*, 2016. 2

[7] Katerina Fragkiadaki, Pulkit Agrawal, Sergey Levine, and Jitendra Malik. Learning visual predictive models of physics for playing billiards. In *International Conference on Learning Representations (ICLR)*, 2016. 2

[8] Radek Grzeszczuk, Demetri Terzopoulos, and Geoffrey Hinton. Neuroanimator: Fast neural network emulation and control of physics-based models. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 9–20. ACM, 1998. 2

[9] Jessica Hamrick, Peter Battaglia, and Joshua B Tenenbaum. Internal physics models guide probabilistic judgments about object dynamics. In *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, 2011. 1

[10] Jessica B Hamrick, Peter W Battaglia, Thomas L Griffiths, and Joshua B Tenenbaum. Inferring mass in complex scenes by mental simulation. *Cognition*, 2016. 1

[11] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *European Conference on Computer Vision (ECCV)*, 2012. 2

[12] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *Neural Information Processing Systems (NIPS)*, 2015. 2, 4.2, 4.3

[13] Adam Lerer, Sam Gross, and Rob Fergus. Learning physical intuition of block towers by example. In *International Conference on Machine Learning (ICML)*, 2016. 1, 2

[14] Wenbin Li, Seyedmajid Azimi, Ales Leonardis, and Mario Fritz. To fall or not to fall: A visual approach to physical stability prediction. *arXiv:1604.00066*, 2016. 2

[15] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *International Conference on Learning Representations (ICLR)*, 2016. 2

[16] Roozbeh Mottaghi, Hessam Bagherinezhad, Mohammad Rastegari, , and Ali Farhadi. Newtonian scene understanding: Unfolding the dynamics of objects in static images. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[17] Roozbeh Mottaghi, Mohammad Rastegari, Abhinav Gupta, and Ali Farhadi. what happens if... learning to predict the effect of forces in images. In *European Conference on Computer Vision (ECCV)*, 2016. 1, 2

[18] Lerrel Pinto, Dhiraj Gandhi, Yuanfeng Han, Yong-Lae Park, , and Abhinav Gupta. The curious robot: Learning visual representations via physical interactions. In *European Conference on Computer Vision (ECCV)*, 2016. 2

[19] Siyuan Qi, Baoxiong Jia, and Song-Chun Zhu. Generalized earley parser: Bridging symbolic grammars and sequence data for future prediction. In *International Conference on Machine Learning (ICML)*, 2018. 2

[20] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. In *International Conference on Machine Learning (ICML)*, 2015. 2

[21] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Neural Information Processing Systems (NIPS)*, 2016. 2

[22] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from variational autoencoders. In *European Conference on*

Computer Vision (ECCV), 2016. 2, 4.1.2

[23] Nicholas Watters, Andrea Tacchetti, Theophane Weber, Razvan Pascanu, Peter Battaglia, and Daniel Zoran. Visual interaction networks. In *Neural Information Processing Systems (NIPS)*, 2017. 2

[24] Jiajun Wu, Ilker Yildirim, Joseph J Lim, William T Freeman, and Joshua B Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In *Neural Information Processing Systems (NIPS)*, 2015. 2

[25] Jiajun Wu, Joseph J Lim, Hongyi Zhang, Joshua B Tenenbaum, and William T Freeman. Physics 101: Learning physical object properties from unlabeled videos. In *BMVC*, 2016. 2

[26] Jiajun Wu, Erika Lu, Pushmeet Kohli, William T. Freeman, and Joshua B. Tenenbaum. Learning to see physics via visual de-animation. In *Neural Information Processing Systems (NIPS)*, 2017. 1, 2

[27] Tianfan Xue, Jiajun Wu, Katherine L. Bouman, and William T. Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Neural Information Processing Systems (NIPS)*, 2016. 2

[28] Renqiao Zhang, Jiajun Wu, Chengkai Zhang, William T Freeman, and Joshua B Tenenbaum. A comparative evaluation of approximate probabilistic simulation and deep neural networks as accounts of human physical scene understanding. In *Proceedings of the 38th AnnualConference of the Cognitive Science Society*, 2016. 2

[29] Bo Zheng, Yibiao Zhao, Joey Yu, Katsushi Ikeuchi, and Song-Chun Zhu. Scene understanding by reasoning stability and safety. *International Journal of Computer Vision (IJCV)*, 2015. 2

[30] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A. Efros. View synthesis by appearance flow. In *European Conference on Computer Vision (ECCV)*, 2016. 2, 4.1.2

[31] Yixin Zhu, Chenfanfu Jiang, Yibiao Zhao, Demetri Terzopoulos, and Song-Chun Zhu. Inferring forces and learning human utilities from videos. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 2