Occlusion resistant learning of intuitive physics from videos

Ronan Riochet, Josef Sivic, Ivan Laptev, Emmanuel Dupoux

Occlusion resistant learning of intuitive physics from videos

Ronan Riochet [1 2] **Josef Sivic** [1 2 3] **Ivan Laptev** [1 2] **Emmanuel Dupoux** [1 2 4]

Abstract

To reach human performance on complex tasks, a key ability for artificial systems is to understand physical interactions between objects, and predict future outcomes of a situation. This ability, often referred to as *intuitive physics*, has recently received attention and several methods were proposed to learn these physical rules from video sequences. Yet, most of these methods are restricted to the case where no, or only limited, occlusions occur. In this work we propose a probabilistic formulation of learning intuitive physics in 3D scenes with significant inter-object occlusions. In our formulation, object positions are modelled as latent variables enabling the reconstruction of the scene. We then propose a series of approximations that make this problem tractable. Object proposals are linked across frames using a combination of a recurrent interaction network, modeling the physics in object space, and a compositional renderer, modeling the way in which objects project onto pixel space. We demonstrate significant improvements over state-of-the-art in the intuitive physics benchmark of Riochet et al. (2018). We apply our method to a second dataset with increasing levels of occlusions, showing it realistically predicts segmentation masks up to 30 frames in the future. Finally, we also show results on predicting motion of objects in real videos.

1. Introduction

Learning intuitive physics has recently raised significant interest in the machine learning literature. To reach human performance on complex visual tasks, artificial systems need to understand the world in terms of macroscopic objects, movements, interactions, etc. Infant development experiments show that young infants quickly acquire an intuitive grasp of how objects interact in the world, and that they use these intuitions for prediction and action planning (Carey, 2009; Baillargeon & Carey, 2012). This includes the notions of gravity (Carey, 2009), continuity of trajectories (Spelke et al., 1995), collisions (Saxe & Carey, 2006), etc. Object permanence, the fact that an object continues to exist when it is occluded, (Kellman & Spelke, 1983), is one of the first concepts developed by infants.

From a modeling point of view, the key scientific question is how to develop general-purpose methods that can make physical predictions in noisy environments, where many variables of the system are unknown. A model that could mimic even some of infant's ability to predict the dynamics of objects and their interactions would be a significant advancement in model-based action planning for robotics (Agrawal et al., 2016; Finn & Levine, 2017). The laws of macroscopic physics are relatively simple and can be readily learned when formulated in 3D cartesian coordinates (Battaglia et al., 2016; Mrowca et al., 2018).

However, learning such laws from real world scenes are difficult for at least two reasons. First, estimating accurate 3D position and velocity of objects is challenging when only their retinal projection is known, even assuming depth information, because partial occlusions by other objects render these positions ambiguous. Second, objects can be fully occluded by other objects for a significant number of frames.

In this paper we address these issues and develop a model for learning intuitive physics in 3D scenes with significant inter-object occlusions. We propose a probabilistic formulation of the intuitive physics problem, whereby object positions are modelled as latent variables enabling the reconstruction of the scene. We then propose a series of approximations that make this problem tractable. In detail, proposals of object positions and velocities (called object states) are derived from object masks, and then linked across frames using a combination of a recurrent interaction network, modeling the physics in object space, and a compositional renderer, modeling the way in which objects project onto pixel space.

Using the proposed approach, we show that it is possible to follow object dynamics in 3D environments with severe inter-object occlusions. We evaluate this ability on the Int-

[1]Ecole Normale Supérieure, CNRS, PSL Research University, Paris, France. [2]Inria, Paris. [3]Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague. [4]Facebook AI Research. Correspondence to: Ronan Riochet <ronan.riochet@polytechnique.edu>.

Phys benchmark (Riochet et al., 2018), a benchmark centered on classifying videos as being physically possible or not. We show better performance compared to (Riochet et al., 2018; Smith et al., 2019). A second set of experiments show that it is possible to learn the physical prediction component of the model even in the presence of severe occlusion, and predict segmentation masks up to 30 frames in the future. Ablation studies and baselines (Battaglia et al., 2016) evaluate the importance of each component of the model, as well the impact of occlusions on performance.

Our model is fully compositional and handles variable number of objects in the scene. Moreover, it does not require as input (or target) annotated inter-frame correspondences during training. Finally, our method still works with no access to ground-truth segmentation, using (noisy) outputs from a pre-trained object/mask detector (He et al., 2018), a first step towards using such models on real videos.

2. Related work

Forward modeling in videos. Forward modeling in videos has been studied for action planning (Ebert et al., 2018; Finn et al., 2016) and as a scheme for unsupervised learning of visual features (Lan et al., 2014; Mathieu et al., 2015). In that setup, a model is given a sequence of frames and generates frames in future time steps (Lan et al., 2014; Mathieu et al., 2015; Finn et al., 2016; Wichers et al., 2018; Zhu et al., 2018; Villegas et al., 2017; Minderer et al., 2019). Such models tend to perform worse when the number of objects increases, sometimes failing to preserve object properties and generating blurry outputs.

Learning dynamics of objects. Longer term predictions can be more successful when done on the level of trajectories of individual objects. For example, in (Wu et al., 2017b), the authors propose "scene de-rendering", a system that builds an object-based, structured representation from a static (synthetic) image. The recovered state can be further used for physical reasoning and future prediction using an off-the-shelf physics engine on both synthetic and real data (Battaglia et al., 2013; Wu et al., 2017a; Smith et al., 2019; Xu et al., 2019). Future prediction from static image is often multi-modal (e.g. car can move forward or backward) and hence models able to predict multiple possible future predictions, e.g. based on variational auto-encoders (Xue et al., 2016), are needed. Autoencoders have been also applied to learn the dynamics of video (Kosiorek et al., 2018; Hsieh et al.) in restricted 2D set-ups and/or with a limited number of objects.

Others have developed structured models that factor object motion and object rendering into two learnable modules. Examples include (Watters et al., 2017; Marco Fraccaro, 2017; Ehrhardt et al., 2017a;b) that combine object-centric dynamic models and visual encoders. Such models parse each frame into a set of object state representations, which are used as input of a "dynamic" model, predicting object motion. However, (Marco Fraccaro, 2017) restrict drastically the complexity of the visual input by working on binary 32x32 frames, and (Ehrhardt et al., 2017a;b; Watters et al., 2017) still need ground truth position of objects as input or target (Watters et al., 2017) for training. However, modeling 3D scenes with significant inter-object occlusions, which is the focus of our work, still remains an open problem.

In our work, we build on learnable models of object dynamics (Battaglia et al., 2016) and (Chang et al., 2016), which have the key property that they are compositional and hence can model a variable number of objects, but extend them to learn from visual input rather than ground truth object state vectors.

Our work is also related to (Janner et al., 2018), who combine an object-centric model of dynamics with a differentiable renderer to predict a single image in a future time, given a single still image as input. In contrast, we develop a probabilistic formulation of intuitive physics that (i) predicts the physical plausibility of an observed dynamic scene, and (ii) infers velocities of objects as latent variables allowing us to predict full trajectories of objects through time despite long complete occlusions. Others have proposed unsupervised methods to discover objects and their interactions in 2D videos (van Steenkiste et al., 2018). It is also possible to construct Hierarchical Relation Networks (Mrowca et al., 2018) or particle-based models (Li et al., 2018), representing objects as graphs and predicting interactions between pairs of objects. However, this task is still challenging and requires full supervision in the form of ground truth position and velocity of objects.

Learning physical properties from visual inputs. Related are also methods for learning physical representations from visual inputs. Examples include (Greff et al., 2019; Burgess et al., 2019) who focus on segmenting images into interpretable objects with disentangled representations. Learning of physical properties, such as mass, volume or coefficients of friction and restitution, has been considered in (Wu et al., 2016). Others have looked at predicting the stability and/or the dynamics of towers of blocks (Lerer et al., 2016; Zhang et al., 2016; Li et al., 2016a;b; Mirza et al., 2017; Groth et al., 2018). Our work is complementary. We don't consider prediction of physical properties but focus on learning models of object dynamics handling inter-object occlusions at both training and test time.

3. Occlusion resistant intuitive physics

This section describes our model for occlusion resistant learning of intuitive physics. In section 3.1 we present an

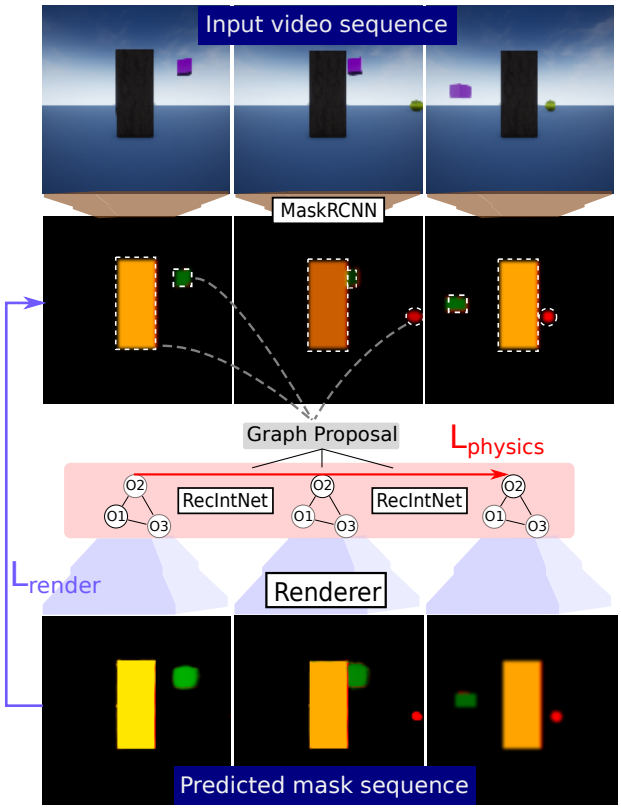

Figure 1. **Overview of our occlusion resistant intuitive physics model.** A pre-trained object detector (MaskRCNN) returns object detections and masks (top). A *graph proposal* matching links object proposals through time: from a pair of frames the Recurrent Interaction Network ($RecIntNet$) predicts next object position and matches it with the closest object proposal. If an object disappears (e.g. due to occlusion - no object proposal), the model keeps the prediction as an *object state*, otherwise this object state is updated with the observation. Finally, the Compositional Rendering Network ($Renderer$) predicts masks from object states and compares them with the observed masks. The errors of predictions of $RecIntNet$ and $Renderer$ on the full sequence are summed into a *physics* and a *render* loss, respectively. The two losses are used to assess whether the observed scene is physically plausibility.

overview of the method, then describe it's two main components: the occlusion-aware compositional renderer that predicts object masks given a scene state representation (section 3.2), and the recurrent interaction network that predicts the scene state evolution over time (section 3.3). Finally, in section 3.4 we describe how these two components are used jointly to decode an entire video clip.

3.1. Intuitive physics via event decoding

We formulate the problem of *event decoding* as that of assigning to a sequence of video frames $F = f_{t=1..T}$ a sequence of underlying object states $S = s_{t=1..T}^{i=1..N}$ that can explain (i.e. reconstruct) this sequence of frames. By object state, we mean object positions, velocities and categories. Within a generative probabilistic model, we therefore try to find the state $\hat{S}$ that maximizes $P(S|F,\theta)$, where θ is a parameter of the model: $\hat{S} = \arg\max_S P(S|F,\theta)$. A nice property of this formulation is that we can use $P(\hat{S}|F,\theta)$ as a measure of the *plausibility* of a video sequence, which is exactly the metric required in the Intphys benchmark.

With Bayes rule, $P(S|F,\theta)$ decomposes into the product of two probabilities that are easier to compute, $P(F|S,\theta)$, the *rendering model*, and $P(S|\theta)$, the *physical model*. This is similar to the decomposition into an acoustic model and a language model in ASR (Jelinek, 1997). The event decoding problem then becomes:

$$\hat{S} = \arg\max_S P(F|S,\theta)P(S|\theta). \tag{1}$$

Such a formulation naturally accounts for occlusion through the rendering model which maps underlying positions into the visible outcome in pixel space. During inference, the physical model is used to fill in the blanks, i.e., imagine what happens behind occluders to maximize the probability of the trajectory. As for the learning problem, it can be formulated as follows:

$$\hat{\theta} = \arg\max_\theta P(F|\theta). \tag{2}$$

In this paper we will apply a number of simplifications to make this problem tractable. First, we operate in mask space and not in pixel space. This is done by using an off-the shelf instance mask detector (Mask-RCNN (He et al., 2018)), making the task of rendering easier, since all of the details and textures are removed from the reconstruction problem. Therefore F is a sequence of (stacks of) binary masks for different objects in the scene. Second, the state space is expressed, not in 3D coordinates, which would require to learn inverse projective geometry, but directly in retinotopic pixel coordinate plus depth (2.5D, something easily available in RGBD cameras). It turns out that learning physics in this space is not more difficult than in the true 3D space. Finally, the probabilistic models are implemented as Neural Networks. The rendering model ($Renderer$) is implemented as a neural network mapping object states into pixel space. The physical model is implemented as a recurrent interaction network ($RecIntNet$), mapping object state at time t as a function of past states.

In practice, computing the $\arg\max$ in eq. (1) is difficult because the states are continuous, the number of objects is unknown, and some objects are occluded in certain frames, yielding a combinatorial explosion regarding how to link hypothetical object states across frames. In this paper, we propose a major approximation to help solving this problem by proceeding in two steps. In the first step, a *scene graph proposal* is computed using bounding boxes to estimate

object position, nearest neighbor matching across nearby frames to estimate velocities, and the roll-out of the physics engine to link the objects across the entire sequence (which is critical to deal with occlusions). The second step consists of optimizing S (given by eq. (1)) by using gradient descent on both models, capitalizing on the fact that both models are differentiable. More precisely, rather than computing probabilities explicitly, we define two losses (that can be interpreted as a proxy for negative log probability): (i) the rendering loss L_{render} that measures the discrepancy between the masks predicted by the renderer and the observed masks in individual frames; and (ii) the physical loss L_{physics} that measures the discrepancy between states predicted by the recurrent interaction network ($RecIntNet$) and the actual observed states. As in ASR, we will combine these two losses with a scaling factor λ, yielding a total loss:

$$\begin{aligned} L_{\text{render}}(S,F) &= \sum_{t=1}^{T} L_{\text{mask}}(Renderer(s_t),F), \\ L_{\text{physics}}(S) &= \sum_{t=1}^{T-1} \|s_{t+1} - RecIntNet(s_t)\|^2, \\ L_{\text{total}}(S,F) &= \lambda L_{\text{render}}(S,F) + (1-\lambda) L_{\text{physics}}(S). \end{aligned} \tag{3}$$

L_{mask} is a pixel-wise loss defined in detail in the supplementary material.

We use the total loss as the objective function to minimize in order to find the interpretation $\hat{S}$ of the masks of a video clip F. And it will be used to provide a plausibility score to decide whether a given scene is physically plausible in the evaluation on the IntPhys Benchmark (section 4.1). As for learning, instead of marginalizing over possible state, we will just optimize the parameters over the point estimate optimal state $\hat{S}$. The aim of this paper is to show that these approximations notwithstanding, a system constructed according to this set-up can yield good results.

3.2. The Compositional Renderer

We introduce a differentiable *Compositional Rendering Network* (or $Renderer$) that predicts a segmentation mask in the image given a list of N objects specified by their x and y position in the image, depth and possibly additional properties such as object type (e.g. sphere, square, ...) or size. Importantly, our neural rendering model has the ability to take a variable number of objects as input and is invariant to the order of objects in the input list. It contains two modules (see Figure 2). First, the *object rendering network* reconstructs a segmentation mask and a depth map for each object. Second, the *occlusion predictor* composes the N predicted object masks into the final scene mask, generating the appropriate pattern of inter-object occlusions obtained from the predicted depth maps of the individual objects.

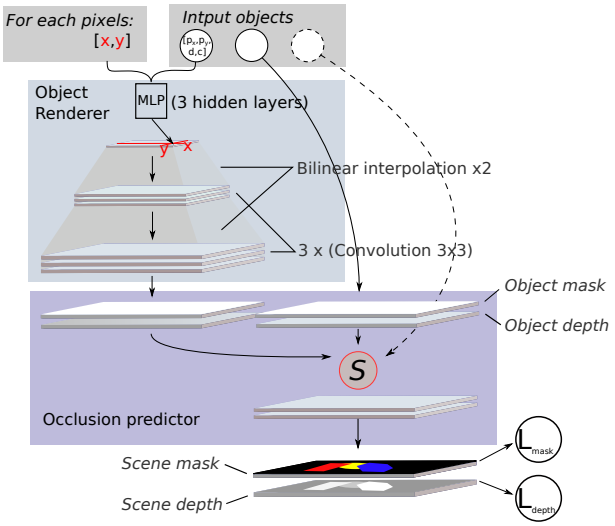

Figure 2. **Compositional Rendering Network** ($Renderer$) Takes as input a list of object states. First, the *object rendering network* reconstructs a segmentation mask and a depth map for each object independently. Second, the *occlusion predictor* composes all predicted object masks into the final scene mask, generating the appropriate pattern of inter-object occlusions obtained from the predicted depth maps of the individual objects.

The Object rendering network takes as input a vector of l values corresponding to the position coordinates (x^k, y^k, d^k) of object k in a frame together with additional dimensions for intrinsic object properties (shape, color and size) $(\boldsymbol{c})$. The network predicts object's binary mask, M^k as well as the depth map D^k. The input vector $(x^k, y^k, d^k, \boldsymbol{c}^k) \in \mathbb{R}^l$ is first copied into a $(l+2) \times 16 \times 16$ tensor, where each 16×16 cell position contains an identical copy of the input vector together with x and y coordinates of the cell. Adding the x and y coordinates may seem redundant, but this kind of *position field* enables a very local computation of the shape of the object and avoids a large number of network parameters (similar architectures were recently also studied in (Liu et al., 2018)).

The input tensor is processed with 1×1 convolution filters. The resulting 16-channel feature map is further processed by three blocks of convolutions. Each block contains three convolutions with filters of size 1×1, 3×3 and 1×1 respectively, and 4, 4 and 16 feature maps, respectively. We use `ReLU` pre-activation before each convolution, and upsample (scale of 2 and bilinear interpolation) feature maps between blocks. The last convolution outputs $N+1$ feature maps of size 128×128, the first feature map encoding depth and the N last feature maps encoding mask predictions for the individual objects. The object rendering network is applied to all objects present, resulting in a set of masks and depth maps denoted as $\{(\hat{M}^k, \hat{D}^k), k = 1..N\}$.

The Occlusion predictor takes as input the masks and depth maps for N objects and aggregates them to construct the final occlusion-consistent mask and depth map. To do so it computes, for each pixel $i, j \leq 128$ and object k the following weight:

$$c^k_{i,j} = \frac{e^{\lambda \hat{D}^k_{i,j}}}{\sum_{q=1}^{N} e^{\lambda \hat{D}^q_{i,j}}}, k = 1..N, \quad (4)$$

where λ is a parameter learned by the model. The final masks and depth maps are computed as a weighted combination of masks $\hat{M}^k_{i,j}$ and depth maps $\hat{D}^k_{i,j}$ for individual objects k: $\hat{M}_{i,j} = \sum_{k=1}^{N} c^k_{i,j} \hat{M}^k_{i,j}$, $\hat{D}_{i,j} = \sum_{k=1}^{N} c^k_{i,j} \hat{D}^k_{i,j}$, where i, j are output pixel coordinates $\forall i, j \leq 128$ and $c^k_{i,j}$ the weights given by (4). The intuition is that the occlusion renderer constructs the final output $(\hat{M}, \hat{D})$ by selecting, for every pixel, the mask with minimal depth (corresponding to the object occluding all other objects). For negative values of λ equation (4) is as a softmin, that selects for every pixel the object with minimal predicted depth. Because λ is a trainable parameter, gradient descent forces it to take large negative values, ensuring good occlusion predictions. Also note that this model does not require to be supervised by the depth field to predict occlusions correctly. In this case, the object rendering network still predicts a feature map $\hat{D}$ that is not equal to the depth anymore but is rather an abstract quantity that preserves the relative order of objects in the view. This allows $Renderer$ to predict occlusions when the target masks are RGB only. However, it still needs depth information in its input (true depth or rank order).

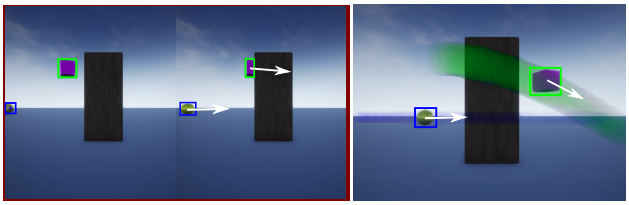

Figure 3. **Illustration of event decoding in the videos of the IntPhys dataset.** A pre-trained object detector returns object proposals in the video (bounding boxes). An initial match is made across two seed neighbouring frames, also estimating object velocity (left, white arrows). The dynamic model (RecIntNet) predicts object positions and velocities in future frames, enabling the match of objects despite significant occlusions (right, bounding box colors and highlights).

3.3. The Recurrent Interaction Network ($RecIntNet$)

To model object dynamics, we build on the Interaction Network (Battaglia et al., 2016), which predicts dynamics of a variable number of objects by modeling their pairwise interactions. Here we describe three extensions of the vanilla Interaction Network model. First, we extend the Interaction Network to model 2.5D scenes where position and velocity have a depth component. Second, we turn the Interaction Network into a recurrent network. Third, we introduce variance in the position predictions, to stabilise the learning phase, and avoid penalizing too much very uncertain predictions. The three extensions are described below.

Modeling compositional object dynamics in 2.5D scenes. As shown in (Battaglia et al., 2016), Interaction Networks can be used to predict object motion both in 3D or in 2D space. Given a list of objects represented by their positions, velocities and size in the Cartesian plane, an Interaction Network models interactions between all pairs of objects, aggregates them over the image and predicts the resulting motion for each object. Here, we model object interactions in 2.5D space, since we have no access to the object position and velocity in the Cartesian space. Instead we have locations and velocities in the image plane plus depth (the distance between the objects and the camera).

Modeling frame sequences. The vanilla Interaction Network (Battaglia et al., 2016) is trained to predict position and velocity of each object in one step into the future. Here, we learn from multiple future frames. We "rollout" the Interaction Network to predict a whole sequence of future states as if a standard Interaction Network was applied in recurrent manner. We found that faster training can be achieved by directly predicting changes in the velocity, hence:

$$[p_1, v_1, c] = [p_0 + \delta t v_0 + \frac{\delta t^2}{2} \mathbf{d_v}, v_0 + \mathbf{d_v}, c], \quad (5)$$

where p_1 and v_1 are position and velocity of the object at time t_1, p_0 and v_0 are position and velocity at time t_0, and $\delta t = t_1 - t_0$ is the time step. Position and velocity in pixel space ($p = [p_x, p_y, d]$ where p_x, p_y are the position of the object in the frame), d is depth and v is the velocity in that space. Hence $\mathbf{d_v}$ can be seen as the *acceleration*, and $(v_0 + \mathbf{d_v})$,$(p_0 + \delta t v_0 + \frac{\delta t^2}{2} \mathbf{d_v})$ as the first and second order Taylor approximations of velocity and position, respectively. Assuming an initial weight distribution close to zero, this gives the model a prior that the object motion is linear.

Prediction uncertainty. To account for prediction uncertainty and stabilize learning, we assume that object position follows a multivariate normal distribution, with diagonal covariance matrix. Each term σ^2_x, σ^2_y, σ^2_d of the covariance matrix represents the uncertainty in prediction, along x-axis, y-axis and depth. Such uncertainty is also given as input to the model, to account for uncertainty either in object detection (first prediction step) or in the recurrent object state prediction. The resulting loss is negative log-likelihood of the target p_1 w.r.t. the multivariate normal distribution, which reduces to:

$$\mathcal{L}\big((\hat{p_1}, \hat{\tau_1}), p_1\big) = \frac{(\hat{p_1} - p_1)^2}{\exp \hat{\tau_1}} + \hat{\tau_1}, \quad (6)$$

where $\hat{\tau_1} = \log \hat{\sigma_1^2}$ is the estimated level of noise propagated through the Recurrent Interaction Network, where σ_1 concatenates σ_x^2, σ_y^2, σ_d^2, p_1 is the ground truth state and $\hat{p_1}$ is the predicted state at time $t+1$. The intuition is that the squared error term in the numerator is weighted by the estimated level of noise $\hat{\tau_1}$, which acts also as an additional regularizer. We found that modeling the prediction uncertainty is important for dealing with longer occlusions, which is the focus of this work.

3.4. Event decoding

Given these components, event decoding is obtained in two steps. First, scene graph proposal gives initial values for object states based on visible objects detected on a frame-by-frame basis. These proposed object states are linked across frames using $RecIntNet$ and a nearest neighbor strategy. Second, this initial proposal of the scene interpretation is then optimized by minimizing the total loss by gradient descent through both $RecIntNet$ and $Renderer$ on the entire sequence of object states, yielding the final interpretation of the scene (example in Figure 3), as well as it's plausibility score (inverse of the total loss). The details of this algorithm are given in the supplementary material.

4. Experiments

In this section we present two sets of experiments evaluating the proposed model. The first set of experiments (section 4.1) is on the IntPhys benchmark that is becoming the de facto standard for evaluating models of intuitive physics[1] (Riochet et al., 2018), and is currently used as evaluation in the DARPA Machine Common Sense program. The second set of experiments (section 4.2) evaluates the accuracy of the predicted object trajectories and is inspired by the evaluation set-up used in (Battaglia et al., 2016) but here done in 3D with inter-object occlusions.

4.1. Evaluation on the IntPhys benchmark

Dataset. The Intphys Benchmark consists in a set of video clips in a virtual environment. Half of the videos depict possible event and half impossible. They are organized in three blocks, each one testing for the ability of artificial systems to discriminate a class of physically impossible events. Block `O1` contains videos where objects may disappear with no reason, thus violating object permanence. In Block `O2`, objects' shape may change during the video, again without any apparent physical reason. In Block `O3`, objects may "jump" from one place to another, thus violating continuity of trajectories. Systems have to provide a plausibility score for each of the 12960 clips and are evaluated in terms of how well they can classify possible and impossible movies. Half of the impossible events (6480 videos) occur in plain sight, and are relatively easy to detect. The other half occurs under complete occlusion, leading to poor performance of current methods (Riochet et al., 2018; Smith et al., 2019).

Along with the test videos, the benchmark contains an additional training set with 15000 videos, with various types of scenes, object movements and textures. Importantly, the training set only consists in possible videos. Solving this task therefore cannot be done by learning a classifier or plausibility score from the training set.

System training. We use the training set to train the Compositional Rendering Network and a MaskRCNN object detector/segmenter from groundtruth object positions and segmentations. We also train the Recurrent Interaction Network to predict trajectories of object 8 frames in the future, given object positions in pairs of input frames. Once trained, we apply the scene graph proposal and optimization algorithm described above and derive the plausibility score which we take as the inverse of a plausibility loss.

Results. Table 1 reports error rates (smaller is better) for the three above mentioned blocks each in the visible and occluded set-up, with "Total" reporting the overall error. We compare performance of our method with two strong baselines Riochet et al. (2018) and the current state-of-the-art on Block `O1` (Smith et al., 2019). We observe a clear improvement over the two other methods, mainly explained by better predictions when impossible events are occluded (see *Occluded* columns). In particular, results in the *Visible* case are rather similar to Riochet et al. (2018), with a slight improvement of 2% on `O1` and 6% on `O3`. On the other hand, improvements on the *Occluded* reach 33% on `O1` and 21% on `O2` clearly demonstrating our model can better deal with occlusions. We could not obtain the Visible/Occluded split score of (Smith et al., 2019) by the time of the submission, thus indicating question marks in the Table 1. On `O3/Occluded`, we observe that our model still struggles to detect correctly impossible events. Interestingly, the same pattern can be observed in human evaluation detailed in Riochet et al. (2018), with a similar error rate in the Mechanical Turk experiment. This tends to show that detecting object "teleportation" under significant occlusions is more complex than other tasks in the benchmark. It would be interesting to confirm this pattern with other methods and/or video stimuli. Overall results demonstrate a clear improvement of our method on the IntPhys benchmark, confirming its ability to follow objects and predict motion under long occlusions.

4.2. Evaluation on Future Prediction

In this section we investigate in more detail the ability of our model to learn to predict future trajectories of objects despite large amounts of inter-object occlusions. We first

[1] www.intphys.com

	Block O1			Block O2			Block O3		
	Visible	Occluded	Total	Visible	Occluded	Total	Visible	Occluded	Total
Ours	**0.05**	**0.19**	**0.12**	**0.11**	**0.31**	**0.21**	**0.26**	**0.47**	**0.37**
(Riochet et al., 2018)	0.07	0.52	0.29	**0.11**	0.52	0.31	0.32	0.51	0.41
(Smith et al., 2019)	-	-	0.27	-	-	-	-	-	-
Human judgement	0.18	0.30	0.24	0.22	0.29	0.25	0.28	0.47	0.37

Table 1. **Results on the IntPhys benchmark.** Relative classification error of our model compared to (Riochet et al., 2018) and (Smith et al., 2019), demonstrating large benefits of our method in scenes with significant occlusions ("Occluded"). Human judgement reports average errors of human judgements, as presented in (Riochet et al., 2018). Lower is better.

describe the dataset and experimental set-up, then discuss the results of object trajectory prediction under varying levels occlusion. Next, we report ablation studies comparing our model with several strong baselines. Finally, we report an experiment demonstrating that our model generalizes to real scenes.

Dataset. We use pybullet[2] physics simulator to generate videos of a variable number of balls of different colors and sizes bouncing in a 3D scene (a large box with solid walls) containing a variable number of smaller static 3D boxes. We generate five datasets, where we vary the camera tilt and the presence of occluders. In the first dataset ("Top view") we record videos with a top camera view (or $90°$), where the borders of the frame coincide with the walls of the box. In the second dataset ("Top view+occ"), we add a large moving object occluding 25% of the scene. Finally, we decrease the camera viewing angle to $45°$, $25°$ and $15°$ degrees, which results in an increasing amount of inter-object object occlusions due to perspective projection of the 3D scene onto a 2D image plane. Contrary to the previous experiment on IntPhys benchmark, we use the ground truth instance masks as the input to our model to remove potential effects due to errors in object detection. Additional details of the datasets and visualizations are given in the supplementary material.

Trajectory prediction in presence of occlusions. In this experiment we initialize the network with the first two frames. We then run a roll-out for N consecutive frames using our model. We consider prediction horizons of 5 and 10 frames, and evaluate the position error as a L2 distance between the predicted and ground truth object positions. L2 distance is computed in the 3D Cartesian scene coordinates so that results are comparable across the different camera tilts. Results are shown in Table 2. We first note that our model (e. RecIntNet) significantly outperforms the linear baseline (a.), which is computed as an extrapolation of the position of objects based on their initial velocities. Moreover, the results of our method are relatively stable across the different challenging setups with occlusions by external objects (Top view+occ) or frequent self-occlusions in tilted views (tilt). This demonstrates the potential ability of our method to be trained from real videos where occlusions usually prevent reliable recovery of object states.

[2] https://pypi.org/project/pybullet

Ablation Studies. As an ablation study we replace the Recurrent Interaction Network ($RecIntNet$) in our model with a multi-layer perceptron (b. MLP baseline in Table 2). This MLP contains four hidden layers of size 180 and is trained the same way as $RecIntNet$, modeling acceleration as described in equation 3.3. To deal with the varying number of objects in the dataset, we pad the inputs with zeros. Comparing the MLP baseline (a.) with our model (e. RecIntNet) we observe that our $RecIntNet$ allows more robust predictions through time.

As a second ablation study, we train the Recurrent Interaction Network without modeling acceleration (eq. 3.3). This is similar to the model described in (Janner et al., 2018), where object representation is not decomposed into position / velocity / intrinsic properties, but is rather a (unstructured) 256-dimensional vector. Results are reported in table 2 (c. NoDyn-RecIntNet). Compared to our full approach (e.), we observe a significant loss in performance, confirming that modeling position and velocity explicitly, and having a constant velocity prior on motion (given by 3.3) improves future predictions.

As a third ablation study, we train a deterministic variant of $RecIntNet$, where only the sequence of states is predicted, without the uncertainty term τ (please see more details in the Supplementary). The loss considered is the mean squared error between the predicted and the observation state. Results are reported in table 2 (d. NoProba-RecIntNet). The results are slightly worse than our model handling uncertainty (d. NoProba-RecIntNet), but close enough to say that this is not a key feature for modeling 5 or 10 frames in the future. In qualitative experiments, however, we observed more robust long-term predictions with uncertainty in our model.

Generalization to real scenes. We test the model trained on top-view synthetic Pybullet videos (without finetuning the weights) on a dataset of 22 real videos containing a variable number of colored balls and blocks in motion recorder with a Microsoft Kinect2 device. Example frames from

	Top view	Top view+occ.	45° tilt	25° tilt	15° tilt
a. Linear baseline	47.6 / 106.0	47.6 /106.0	47.6 / 106.0	47.6 / 106.0	47.6 / 106.0
b. MLP baseline	13.1 / 15.7	17.3 / 19.2	18.1 / 23.8	17.6 / 24.6	19.4 / 26.2
c. NoDyn-RecIntNet	21.2 / 46.2	23.7 / 46.7	22.5 / 42.8	23.1 / 43.3	24.9 / 44.4
d. NoProba-RecIntNet	6.3 / 11.5	12.4 / 14.7	**8.0** / 15.9	8.12 / 16.3	11.2 / 19.6
e. RecIntNet (Ours)	6.3 / **9.2**	**11.7 / 13.5**	8.01 / **14.5**	**8.1 / 15.0**	11.2 / **18.1**

Table 2. **Object trajectory prediction in the synthetic dataset.** Average Euclidean L2 distance in pixels between predicted and ground truth positions, for a prediction horizon of 5 / 10 frames (lower is better). To compute the distance, the pixel-based x-y-d coordinates of objects are projected back in an untilted 200x200x200 reference Cartesian coordinate system.

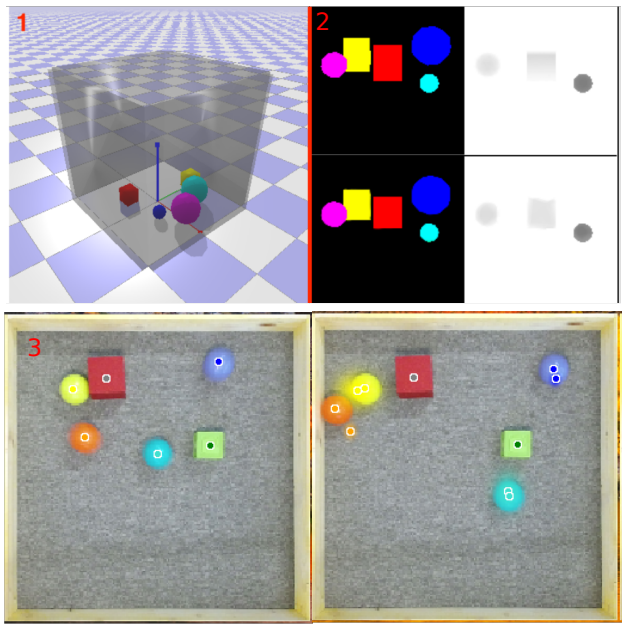

Figure 4. **Images from the Future Prediction experiment** 1: An overview of the pybullet scene. 2: Sample video frames (instance mask + depth field) from our datasets (top) together with predictions obtained by our model (bottom), taken from from the tilted 25° experiments. 3: example of prediction for a real video, with a prediction span of 8 frames. The small colored dots show the predicted positions of objects together with the estimated uncertainty shown by the colored "cloud". The same colored dot is also shown in the (ground truth) center of each object. The prediction is correct when the two dots coincide. (see additional videos).

the data are shown in figure 4. Results are reported in the supplementary and demonstrate that our model generalizes to real data and show clear improvements over the linear and MLP baselines.

Additional results in the supplementary material. In addition to the forward prediction, we evaluate our method on the task of following objects in the scene. Details and results can be found in the supplementary material (section 5).

5. Discussion

Learning the physics of simple macroscopic object dynamics and interactions is a relatively easy task when ground truth coordinates are provided to the system, and techniques like Interaction Networks trained with a future frame prediction loss are quite successful (Battaglia et al., 2016; Mrowca et al., 2018). In real-life applications, the physical state of objects is not available and has to be inferred from sensors. In such case inter-object occlusions make these observations noisy and sometimes missing.

Here we present a probabilistic formulation of the intuitive physics problem, where observations are noisy and the goal is to infer the most likely underlying object states. This physical state is the solution of an optimization problem involving i) a physics loss: objects states should be coherent in time, and ii) a render loss: the resulting scene at a given time should match with the observed frame. We present a method to find an approximate solution to this problem, that is compositional (does not restrict the number of objects) and handles occlusions. We show its ability to learn object dynamics and demonstrate it outperforms existing methods on the intuitive physics benchmark IntPhys.

A second set of experiments studies the impact of occlusions on intuitive physics learning. During training, occlusions act like missing data because the object position is not available to the model. However, we found that it is possible to learn good models compared to baselines, even in challenging scenes with significant inter-object occlusions. We also notice that projective geometry is not, in and of itself, a difficulty in the process. Indeed, when an our dynamics model is fed, not with 3D Cartesian object coordinates, but with a 2.5D projective referential such as the xy position of objects in a retina (plus depth), the accuracy of the prediction remains unchanged compared with the Cartesian ground truth. Outcomes of these experiments can be seen in the google drive (link). This work, along with recent improvement of object segmentation models (He et al., 2018) put a first step towards learning intuitive physics from real videos.

Further work needs to be done to fully train this system end-to-end, in particular, by learning the renderer and the interaction network jointly. This could be done within our probabilistic framework by improving the initialization step

of our system (scene graph proposal). Instead of using a relatively simple heuristics yielding a single proposal per video clip, one could generate multiple proposals (a decoding lattice) that would be reranked with the plausibility loss. This would enable more robust joint learning by marginalizing over alternative event graphs instead of using a single point estimate as we do here. Finally object segmentation itself could be learned jointly, as this would allow exploiting physical regularities of the visual world as a bootstrap to learn better visual representations.

6. Acknowledgements

We would like to thank Malo Huard for his help in implementing the renderer, and anonymous reviewers for their helpful comments.

This work was partly supported by the European Regional Development Fund under the project IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15_003/0000468) and by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

References

Agrawal, P., Nair, A., Abbeel, P., Malik, J., and Levine, S. Learning to Poke by Poking: Experiential Learning of Intuitive Physics. *CoRR*, abs/1606.07419, 2016. URL http://arxiv.org/abs/1606.07419.

Baillargeon, R. and Carey, S. Core Cognition and Beyond. In Pauen, S. (ed.), *Early childhood development and later outcome*, pp. 33–65. Cambridge University Press, New York, 2012.

Battaglia, P., Pascanu, R., Lai, M., Rezende, D. J., and others. Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems*, pp. 4502–4510, 2016.

Battaglia, P. W., Hamrick, J. B., and Tenenbaum, J. B. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences of the United States of America*, 110(45), 2013. URL http://www.pnas.org/content/110/45/18327.

Burgess, C. P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M., and Lerchner, A. Monet: Unsupervised scene decomposition and representation. *CoRR*, abs/1901.11390, 2019. URL http://arxiv.org/abs/1901.11390.

Carey, S. *The origin of concepts*. Oxford series in cognitive development. Oxford University Press, Oxford ; New York, 2009. ISBN 978-0-19-536763-8 0-19-536763-4.

Chang, M. B., Ullman, T., Torralba, A., and Tenenbaum, J. B. A compositional object-based approach to learning physical dynamics. *arXiv preprint arXiv:1612.00341*, 2016.

Ebert, F., Finn, C., Dasari, S., Xie, A., Lee, A., and Levine, S. Visual Foresight: Model-Based Deep Reinforcement Learning for Vision-Based Robotic Control. *arXiv:1812.00568 [cs]*, December 2018. URL http://arxiv.org/abs/1812.00568. arXiv: 1812.00568.

Ehrhardt, S., Monszpart, A., Mitra, N. J., and Vedaldi, A. Learning A Physical Long-term Predictor. *CoRR*, abs/1703.00247, 2017a.

Ehrhardt, S., Monszpart, A., Mitra, N. J., and Vedaldi, A. Taking Visual Motion Prediction To New Heightfields. 2017b.

Finn, C. and Levine, S. Deep visual foresight for planning robot motion. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 2786–2793. IEEE, 2017.

Finn, C., Goodfellow, I., and Levine, S. Unsupervised learning for physical interaction through video prediction. In *Advances in neural information processing systems*, pp. 64–72, 2016.

Greff, K., Kaufman, R. L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M., and Lerchner, A. Multi-Object Representation Learning with Iterative Variational Inference. *arXiv:1903.00450 [cs, stat]*, March 2019. URL http://arxiv.org/abs/1903.00450. arXiv: 1903.00450.

Groth, O., Fuchs, F. B., Posner, I., and Vedaldi, A. Shapestacks: Learning vision-based physical intuition for generalised object stacking, 2018.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask R-CNN. *arXiv:1703.06870 [cs]*, January 2018. URL http://arxiv.org/abs/1703.06870. arXiv: 1703.06870.

Hsieh, J.-T., Liu, B., Huang, D.-A., Fei-Fei, L. F., and Niebles, J. C. Learning to Decompose and Disentangle Representations for Video Prediction. pp. 10.

Janner, M., Levine, S., Freeman, W. T., Tenenbaum, J. B., Finn, C., and Wu, J. Reasoning About Physical Interactions with Object-Oriented Prediction and Planning. *arXiv:1812.10972 [cs, stat]*, December 2018. URL http://arxiv.org/abs/1812.10972. arXiv: 1812.10972.

Jelinek, F. *Statistical methods for speech recognition*. MIT press, 1997.

Kellman, P. J. and Spelke, E. S. Perception of partly occluded objects in infancy. *Cognitive psychology*, 15(4): 483–524, 1983.

Kosiorek, A., Kim, H., Teh, Y. W., and Posner, I. Sequential Attend, Infer, Repeat: Generative Modelling of Moving Objects. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 8606–8616. Curran Associates, Inc., 2018.

Lan, T., Chen, T.-C., and Savarese, S. A hierarchical representation for future action prediction. In *European Conference on Computer Vision*, pp. 689–704. Springer, 2014.

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

Li, W., Leonardis, A., and Fritz, M. To Fall Or Not To Fall: A Visual Approach to Physical Stability Prediction. *arXiv preprint*, 2016a. URL https://arxiv.org/abs/1604.00066.

Li, W., Leonardis, A., and Fritz, M. Visual stability prediction and its application to manipulation. *arXiv preprint arXiv:1609.04861*, 2016b.

Li, Y., Wu, J., Tedrake, R., Tenenbaum, J. B., and Torralba, A. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. *CoRR*, abs/1810.01566, 2018. URL http://arxiv.org/abs/1810.01566.

Liu, R., Lehman, J., Molino, P., Such, F. P., Frank, E., Sergeev, A., and Yosinski, J. An intriguing failing of convolutional neural networks and the coordconv solution. *CoRR*, abs/1807.03247, 2018. URL http://arxiv.org/abs/1807.03247.

Marco Fraccaro, Simon Kamronn, U. P. O. W. A Disentangled Recognition and Nonlinear Dynamics Model for Unsupervised Learning. *Advances in Neural Information Processing Systems 30, NIPS*, 2017.

Mathieu, M., Couprie, C., and LeCun, Y. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

Minderer, M., Sun, C., Villegas, R., Cole, F., Murphy, K., and Lee, H. Unsupervised learning of object structure and dynamics from videos. *CoRR*, abs/1906.07889, 2019. URL http://arxiv.org/abs/1906.07889.

Mirza, M., Courville, A., and Bengio, Y. Generalizable features from unsupervised learning. *ICLR Workshop submission*, 2017. URL https://openreview.net/pdf?id=BynzZolYg.

Mrowca, D., Zhuang, C., Wang, E., Haber, N., Fei-Fei, L. F., Tenenbaum, J., and Yamins, D. L. Flexible neural representation for physics prediction. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 8813–8824. Curran Associates, Inc., 2018.

Riochet, R., Ynocente Castro, M., Bernard, M., Lerer, A., Fergus, R., Izard, V., and Dupoux, E. IntPhys: A Framework and Benchmark for Visual Intuitive Physics Reasoning. *ArXiv e-prints*, March 2018.

Saxe, R. and Carey, S. The perception of causality in infancy. *Acta psychologica*, 123(1):144–165, 2006.

Smith, K. A., Mei, L., Yao, S., jun Wu, J., Spelke, E. S., Tenenbaum, J., and Ullman, T. Modeling expectation violation in intuitive physics with coarse probabilistic object representations. In *NeurIPS*, 2019.

Spelke, E. S., Kestenbaum, R., Simons, D. J., and Wein, D. Spatiotemporal continuity, smoothness of motion and object identity in infancy. *British Journal of Developmental Psychology*, 13(2):113–142, 1995.

van Steenkiste, S., Chang, M., Greff, K., and Schmidhuber, J. Relational Neural Expectation Maximization: Unsupervised Discovery of Objects and their Interactions. *arXiv:1802.10353 [cs]*, February 2018. URL http://arxiv.org/abs/1802.10353. arXiv: 1802.10353.

Villegas, R., Yang, J., Hong, S., Lin, X., and Lee, H. Decomposing motion and content for natural video sequence prediction. *CoRR*, abs/1706.08033, 2017. URL http://arxiv.org/abs/1706.08033.

Watters, N., Tacchetti, A., Weber, T., Pascanu, R., Battaglia, P., and Zoran, D. Visual Interaction Networks. *CoRR*, abs/1706.01433, 2017. URL http://arxiv.org/abs/1706.01433.

Wichers, N., Villegas, R., Erhan, D., and Lee, H. Hierarchical long-term video prediction without supervision. *CoRR*, abs/1806.04768, 2018. URL http://arxiv.org/abs/1806.04768.

Wu, J., Lim, J. J., Zhang, H., Tenenbaum, J. B., and Freeman, W. T. Physics 101: Learning Physical Object Properties from Unlabeled Videos. In *BMVC*, 2016.

Wu, J., Lu, E., Kohli, P., Freeman, B., and Tenenbaum, J. Learning to see physics via visual de-animation. In *Advances in Neural Information Processing Systems*, pp. 152–163, 2017a.

Wu, J., Tenenbaum, J. B., and Kohli, P. Neural scene de-rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 699–707, 2017b.

Xu, Z., Wu, J., Zeng, A., Tenenbaum, J. B., and Song, S. Densephysnet: Learning dense physical object representations via multi-step dynamic interactions. *CoRR*, abs/1906.03853, 2019. URL http://arxiv.org/abs/1906.03853.

Xue, T., Wu, J., Bouman, K., and Freeman, B. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Advances in Neural Information Processing Systems*, pp. 91–99, 2016.

Zhang, R., Wu, J., Zhang, C., Freeman, W. T., and Tenenbaum, J. B. A Comparative Evaluation of Approximate Probabilistic Simulation and Deep Neural Networks as Accounts of Human Physical Scene Understanding. *CogSci*, 2016. URL http://blocks.csail.mit.edu/.

Zhu, G., Huang, Z., and Zhang, C. Object-Oriented Dynamics Predictor. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 9826–9837. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/8187-object-oriented-dynamics-predictor.pdf.

Supplementary material

This supplementary material: (i) describes the provided supplementary videos (section 7), (ii) provides additional training details (section 8), (iii) explains in more depth the *event decoding* procedure defined in section 3.4 in the main paper (section 9) (iv) gives details of the datasets used in the subsection 4.2 (section 10), (v) provides additional ablation studies and comparisons with baselines (sections 11, 12, 13, 14).

7. Description of supplementary videos

In this section we present qualitative results of our method on different datasets. We first show videos from IntPhys benchmark, where inferred object states are depicted onto observed frames. Then we show differents outputs on the pybullet datasets, for different levels of occlusions. Finally we present examples of predictions from our Recurrent Interaction Network on real scenes.

The videos are in the google drive: `https://drive.google.com/open?id=1Qc8flIAxUGzfRfeFyyUEGXe6J5AUGUjE` in the `videos/` subdirectory. Please see also the `README` slideshow in the same directory.

7.1. IntPhys benchmark

The Intphys Benchmark consists in a set of video clips in a virtual environment. Half of the videos are possible event and half are impossible, the goal being to discriminate the two.

In the following we show impossible events, along with outputs of our event decoding method. Our dynamics and rendering models predict future frames (masks) in the videos, which are compared with the observed masks (pre-trained detector). This allows us to derive a plausibility loss used to discriminate possible and impossible events (see section 4.1).

- **occluded_impossible_*.mp4** show examples of impossible videos from the IntPhys benchmark, along with visualization of our method. Each video contains four splits; on top/left is shown the raw input frame; on bottom/left is the mask obtained from the raw frame with the pre-trained mask detector (which we call *observed mask*); on top/right is the raw frame with superimposed output physical states predicted by our method; on bottom/right is the reconstructed mask obtained with the Compositional Renderer (which we call *predicted mask*). Throughout the sequence, our method predicts the full trajectory of objects. When an object should be visible (i.e. not behind an occluder), the renderer predicts correctly its mask. If at the same time the object has disappeared from the observed mask, or changed too much in position or shape, it causes a mismatch between the predicted and the observed masks, hence a higher plausibility loss. This plausibility loss is use for the classification task of IntPhys benchmark (see quantitative results in main paper, section 4.1).

- **visible_impossible_*.mp4** show similar videos but with impossible events occurring in the "visible" (easier) task of the IntPhys benchmark.

- **intphys_*.mp4** show object following in the IntPhys training set.

7.2. Pybullet experiments

We present qualitative results on our Pybullet dataset. We construct videos including a various number of objects with different points of view and increasing levels of camera tilts introducing inter-object occlusions. First, we show predicted physical states drawn on object states, to demonstrate the ability of the method to track objects under occlusions. Then we show videos of long rollouts where, from one pair of input frames, we predict a full trajectory and render masks with the Compositional Neural Renderer.

- **scene_overview.mp4** shows raw videos of the entire environment.

- **tracking_occlusions_*.mp4** show examples of position prediction through complete occlusions, using our event decoding procedure. This shows that our model can keep track of the object identity through complete occlusions, mimicking “object permanence".

- **one_class*.mp4** show different examples of our model following motion of multiple objects in the scene. All balls have the same color which makes them difficult to follow in case of mutual interactions. Videos come

from tilted 25° experiments, which are the most challenging because they include inter-object occlusions. Dots represent the predicted position of each object, the color being its identity. Our model shows very good predictions with small colored markers (dots) well centered in the middle of each object, with marker color remaining constant for each object preserving the object identity during occlusions and collisions. **one_class_raw*.mp4** show rendered original views of the same dynamic scenes but imaged from a different viewpoint for better understanding.

- **rollout_0.mp4**, **rollout_1.mp4** show three different prediction roll-outs of the Recurrent Interaction Network (without event decoding procedure). From left to right: ground truth trajectories, our model trained of state, our model trained on masks, our model trained on masks with occlusions during training. Rollout length is 20 frames.

- **rollout_tilt*_model.mp4** and **rollout_tilt*_groundtruth.mp4** show the same dynamic scene but observed with various camera tilts (e.g. **tilt45_model.mp4** show a video for a camera tilt of 45 degrees). ***_model.mp4** are predicted roll-outs of our Recurrent Interaction Network ($RecIntNet$), without event decoding. ***_groundtruth.mp4** are the corresponding ground-truth trajectories, rendered with the *Compositional Rendering Network*.

- **rollout_pybullet_*.mp4** show free roll-outs (no event decoding) on synthetic dataset.

7.3. Real videos

- **rollout_real_*.mp4** show generalization to real scenes.

8. Training details

This section gives details of the offline pre-training of the Compositional Rendering Network and detailed outline of the algorithm for training the Recurrent Interaction Network.

Pre-Training the Compositional Rendering Network. We train the neural renderer to predict mask and depth $\hat{M}_t, \hat{D}_t$ from a list of objects $[p_x, p_y, d, \boldsymbol{c}]$ where p_x, p_y are x-y coordinates of the object in the frame, d is the distance between the object and the camera and $\boldsymbol{c}$ is a vector for intrinsic object properties containing the size of the object, its class (in our experiments a binary variable for whether the object is a ball, a square or an occluder) and its color as vector in $[0, 1]^3$. In IntPhys benchmark, occluders are not modeled with a single point $[p_x, p_y, d, \boldsymbol{c}]$ but with four points $[p_x^k, p_y^k, d^k], k = 1..4$ corresponding to the four corners of the quadrilateral. These four corners are computed from the occluder instance mask, after detecting contours and applying Ramer–Douglas–Peucker algorithm to approximate the shape with a quadrilateral.

The target mask is a 128×128 image where each pixel value indicates the index of the corresponding object mask (0 for the background, $i \in 1..N$ for objects). The loss on the mask is negative log-likelihood, which corresponds to the average classification loss on each pixel

$$L_{\text{mask}}(\hat{M}, M) = \sum_{i \leq h, j \leq w} \sum_{n \leq N} \mathbf{1}(M_{i,j} = n) log(\hat{M}_{i,j,n}), \tag{7}$$

where the first sum is over individual pixels indexed by i and j, the second sum is over the individual objects indexed by n, $\forall \hat{M} \in [0, 1]^{h \times w \times N}$ are the predicted (soft-) object masks, and $\forall M \in [\![1, N]\!]^{h \times w}$ is the scene ground truth mask containing all objects.

The target depth map is a 128×128 image with values being normalized to the [-1,1] interval during training. The loss on the depth map prediction is the mean squared error

$$L_{\text{depth}}(\hat{D}, D) = \sum_{i \leq h, j \leq w} (\hat{D}_{i,j} - D_{i,j})^2, \tag{8}$$

where $\forall \hat{D}$ and $D \in \mathbb{R}^{h \times w}$ are the predicted and ground truth depth maps, respectively. The final loss used to train the renderer is the weighted sum of losses on masks and depth maps, $L = 0.7 * L_{mask} + 0.3 * L_{depth}$. We use the `Adam` optimizer with default parameters, and reduce learning rate by a factor 10 each time the loss on the validation set does not decrease during 10 epochs. We pre-train the network on a separate set of 15000 images generated with `pybullet` and containing similar objects as in our videos.

Training details of the Recurrent Interaction Network. From a sequence of L frames with their instance masks we compute objects position, size and shape (see section 3.2 in the main body). Initial velocities of objects are estimated as the position deltas between the first two positions. This initial state (position, velocity, size and shape of all objects) is given as input of the Recurrent Interaction Network to predict the next L-2 states. The predicted L-2 positions are compared with observed object positions. The sum of prediction errors (section 3.3 in core paper) is used as loss to train parameters of the Recurrent Interaction Network. Optimization is done via gradient descent, using Adam with learning rate $1e-3$, reducing learning by a factor of 10 each time loss on validation plateaus during 10 epochs. We tried several sequence lengths (4, 6, 10), 10 giving the most stable results. During such sequence, when an object was occluded (thus position not being observed), we set its loss to zero.

9. Event Decoding

The detailed outline of the event decoding procedure described in section 9 of the main paper is given in Algorithm 1. Two example figures (Figure 5 & 6) gives an intuition behind the *render* and *physics* losses.

10. Datasets

To validate our model, we use pybullet[1] physics simulator to generate videos of variable number of balls of different colors and sizes bouncing in a 3D scene (a large box with solid walls) containing a variable number of smaller static 3D boxes. We generate five dataset versions, where we vary the camera tilt and the presence of occluders. All experiments are made with datasets of 12,000 videos of 30 frames (with a frame rate of 20 frames per second). For each dataset, we keep 2,000 videos separate to pre-train the renderer, $9,000$ videos to train the physics predictor and $1,000$ videos for evaluation. Our scene contains a variable number of balls (up to 6) with random initial positions and velocities, bouncing against each other and the walls. Initial positions are sampled from a uniform distribution in the box $[1, 200]^2$, all balls lying on the ground. Initial velocities along x and y axes are sampled in *Unif*$([-25, 25])$ units per frame, initial velocity along z-axis is set to 0. The radius of each ball is sampled uniformly in $[10, 40]$. Scenes also contain a variable number of boxes (up to 2) fixed to the floor, against which balls can collide. Contrary to (Battaglia et al., 2016) where authors set a frame rate of 1000 frames per second, we sample 30 frames per second, which is more reasonable when working with masks (because of the computation cost of mask prediction).

Top-view. In the first dataset we record videos with a top camera view, where the borders of the frame coincide with the walls of the box. Here, initial motion is orthogonal to the camera, which makes this dataset very similar to the 2D bouncing balls datasets presented in (Battaglia et al., 2016) and (Watters et al., 2017). However, our dataset is 3D and because of collisions and the fact that the balls have different sizes, balls can jump on top of each other, making occlusions possible, even if not frequent.

Top-view with Occlusions. To test the ability of our method to learn object dynamics in environments where occlusions occur frequently, we record the second dataset including frequent occlusions. We add an occluder to the scene, which is an object of irregular shape (an airplane), occluding 25% of the frame and moving in 3D between the balls and the camera. This occluder has a rectilinear motion and goes from the bottom to the top of the frame during the whole video sequence. Sample frames and rendered

[1] https://pypi.org/project/pybullet

Algorithm 1 Event decoding procedure

Data:
- T: length of the video
- f_t, m_t $t = 1..T$: videos frames, segmentation masks
- `Detection(`m_t`)`: returns centroid and size of instance masks
- `RecIntNet`: Pre-trained Recurrent Interaction Network
- `Rend`: Pre-trained Neural Renderer
- `ClosestMatch(a,b)`: for a, b two lists of objects, computes the optimal ordering of elements in b to match those in a
- $0 < \lambda < 1$: weighting physical and visual losses

Result:
- Estimated states $s_{1...T}$
- Plausibility loss for the video

Initialization:
$d_{t=1..T} = \text{Detection}(f_t)$ $n_t \leftarrow (\#\{d_t\}, \text{mean}_t size(d_t))$
$t^* \leftarrow \arg\max_t(n_t + n_{t+1})$
`//`(t^*, t^*+1) `is the pair of frames containing the maximum number of objects (with the max number of visible pixels in case of equality).`
$(p_{t^*}, p_{t^*+1}) \leftarrow (d_{t^*}, \text{ClosestMatch}(d_{t^*}, d_{t^*+1}))$
`//Rearange` d_{t^*+1} `to have same object ordering as in` d_{t^*}.

Graph Proposal:
`//`t^* `is a good starting point for parsing the scene (because we observe most of the objects during two consecutive frames). We use RecIntNet to predict the next position of each object, which we link to an object detection. Repeating this step until the end of the video returns object trajectory candidates.`
$v_{t^*+1} \leftarrow p_{t^*+1} - p_{t^*}$
$s_{t^*+1} \leftarrow [p_{t^*+1}, v_{t^*+1}]$
for $t \in \{t^*+1, .., T\}$ **do**
  $\hat{s}_{t+1} \leftarrow \text{RecIntNet}(s_t)$
  $s_{t+1} \leftarrow \text{ClosestMatch}(\hat{s}_{t+1}, d_{t+1})$
`//Backward: do the same from` t^* `to 1.`

Differentiable optimization:
`//`$\hat{s}_{t=1..T}$ `is a sequence of physical states. At every time step` t `it contains the position, velocity, size and shape of all objects, in the same order. Due to occlusions and detection errors, it is sub-optimal and can be refined to minimize equation 3 in the main paper.`
$\text{Loss}_{\text{physics}}(s) \leftarrow \sum_{t=1}^{T} \|\hat{s}_{t+1} - s_{t+1}\|^2$
$\text{Loss}_{\text{visual}}(s) \leftarrow \sum_{t=1}^{T} \text{NLL}(\text{Rend}(s_t), m_t)$
$\text{Loss}_{\text{plausibility}}(s) \leftarrow \lambda \text{Loss}_{\text{physics}}(s) + (1 - \lambda)\text{Loss}_{\text{visual}}(s)$
(Estimated states, plausibility loss) $\leftarrow \text{SGD}_s(\text{Loss}_{\text{plausibility}}(s))$
`//with` $lr = 1e-3$ `and` $n_{\text{steps}} = 1000$

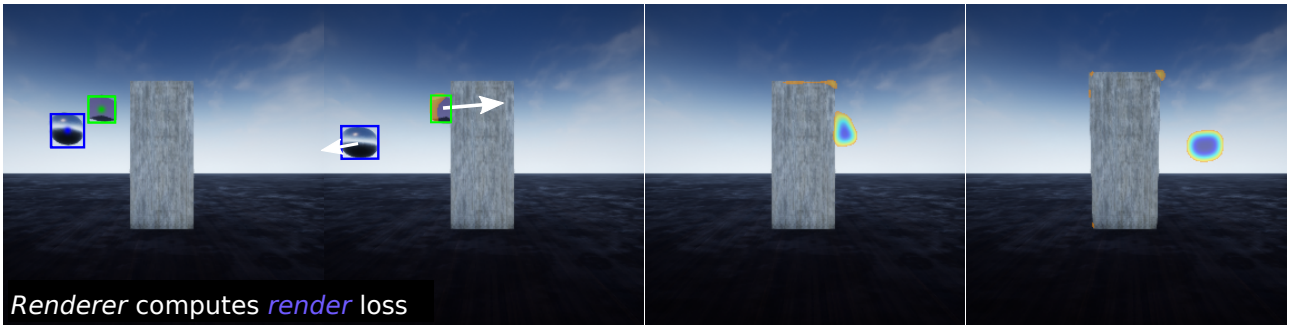

Figure 5. **Video example from the IntPhys benchmark.** Four frames from a video in block O1, with superimposed heatmaps. Heatmaps (colored blobs) correspond to the difference, per pixel, between the predicted and the observed object mask. In these video, a cube moves from left to right but disappears behind the occluder. The Recurrent Interaction Network predicts correctly its motion behind the occluder and the Compositional Renderer reconstructs its mask. The fact that the object is absent in the observed mask leads to a large *render* loss, illustrated by the high heatmap values (violet) at the position where the ball is expected to be.

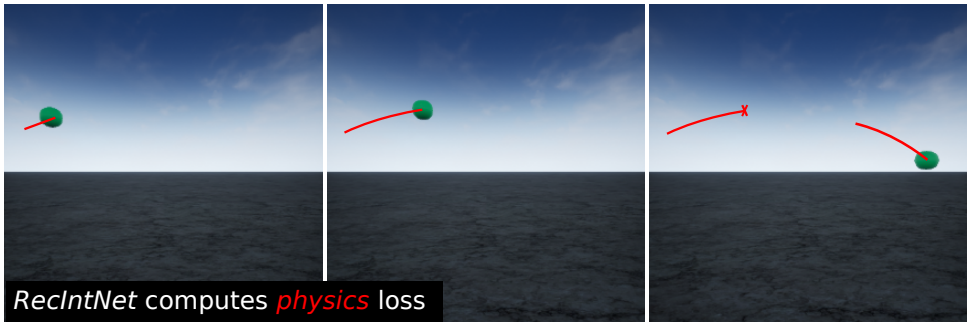

Figure 6. **Video example from the IntPhys benchmark.** Three frames from a video in block O2, where an object "jumps" from one place to another. The graph proposal phase returns the right trajectory of the object but the Recurrent Interaction Network returns a high *physics* loss at the moment of the jump, because the observed position is far from the predicted one.

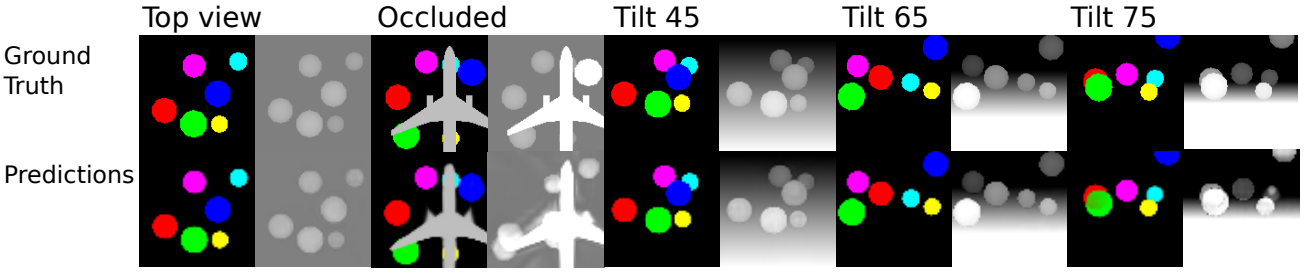

Figure 7. Sample video frames (instance mask + depth field) from our datasets (top) together with predictions obtained by our model (bottom). Taken from the top-view, occluded and tilted experiments. **Please see additional video results in the google drive https://drive.google.com/open?id=1Qc8flIAxUGzfRfeFyyUEGXe6J5AUGUjE.**

predictions can be found in the supplementary material.

Tilted-views. In three additional datasets we keep the same objects and motions but tilt the camera with angles of $45°$, $65°$ and $75°$ degrees. Increasing the tilt of the camera results in more severe inter-object occlusions (both partial and complete) where the balls pass in front of each other, and in front and behind the static boxes, at different distances to the camera. In addition, the ball trajectories are becoming more complex due to increasing perspective effects. In contrary to the top-view experiment, the motion is not orthogonal to the camera plane anymore, and depth becomes crucial to predict the future motion.

11. Ablation studies

For the purpose of comparison, we also evaluate three models trained using ground truth object states. Results are shown in table . Our Recurrent Interaction Network trained on ground truth object states gives similar results to the model of (Battaglia et al., 2016). As expected, training on ground truth states (effectively ignoring occlusions and other effects) performs better than training from object masks and depth.

	Top view	45° tilt	25° tilt	15° tilt
NoProba-RIN	4.76 / 9.72	6.21 / 10.0	5.2 / 12.2	7.3 / 13.8
RIN	4.5 / 9.0	6.0 / 9.6	5.2 / 12.2	7.3 / 13.2
2016**	3.6 / 10.1	4.5 / 9.9	4.5 / 11.0	5.3 / 12.3

Table 3. Average Euclidean (L2) distance (in an untilted 200x200x200 reference Cartesian coordinate system) between predicted and ground truth positions, for a prediction horizon of 5 frames / 10 frames, trained on ground truth positions. **(Battaglia et al., 2016) is trained with more supervision, since target values include ground truth velocities, not available to other methods.

12. Roll-out results

We evaluate our model on *object following*, applying an online variant of the scene decoding procedure detailed in 9. This online variant consists in applying the state optimization sequentially (as new frames arrive), instead of on the full sequence. For each new frame, the state prediction $\hat{s}_{t+1}$ given by $RecIntNet$ is used to predict a resulting mask. This mask is compared to the observation, and we apply directly the final step in Algorithm 1 (Differentiable optimization). It consists in minimizing $\lambda \text{Loss}_{\text{physics}}(s) + (1-\lambda)\text{Loss}_{\text{visual}}(s)$ via gradient descent over the state s. During full occlusion, the position is solely determined by $RecIntNet$, since $\text{Loss}_{\text{render}}$ has a zero gradient. When the object is completely or partially visible, the $\text{Loss}_{\text{render}}$ in the minimization make the predicted state closer to its observed value. To test object following, we measure the accuracy of the position estimates across long sequences (up to 30 frames) containing occlusions. Table 4 shows the percentage of object predictions that diverge by more than an object diameter (20 pixels) using this method. The performance is very good, even for tilted views. In Figure 8, we report the proportion of correctly followed objects for different rollout lengths (5, 10 and 30 frames) as a function of the distance error (pixels). Note that the size of the smallest object is around 20 pixels.

13. Experiment with real videos

We construct a dataset of 22 real videos, containing a variable number of colored balls and blocks in motion. Videos

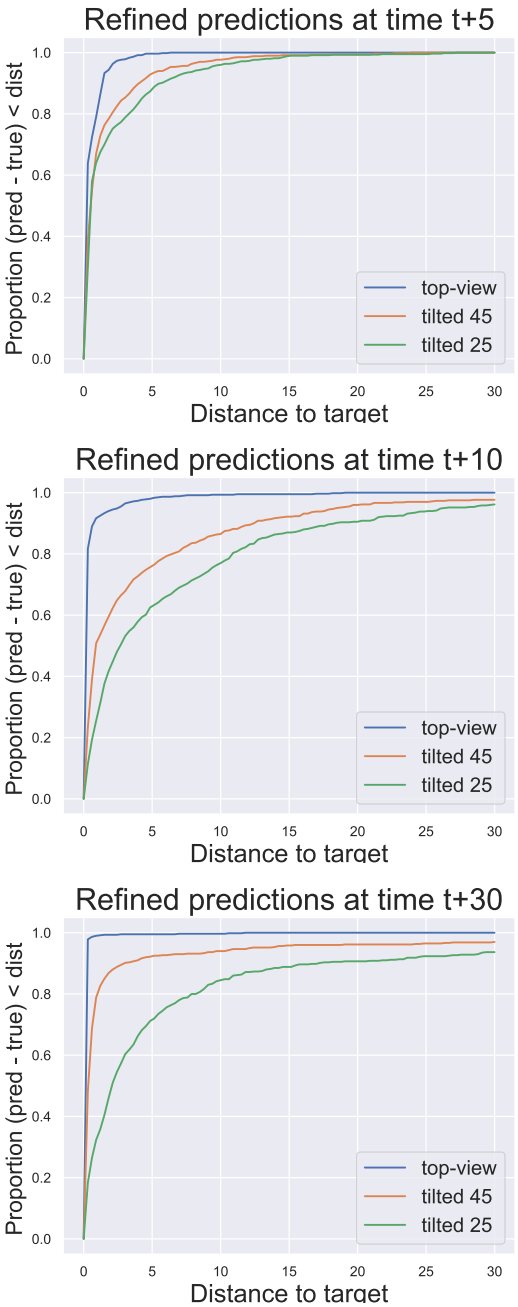

Figure 8. Proportion of correctly followed objects (y-axis) as a function of the distance error in pixels (x-axis) for our approach using online event decoding. The different plots correspond to roll-out lengths of 5 (left), 10 (middle) and 30 (right) frames. Different curves correspond to different camera view angles (top-view, tilted 45 degrees and tilted 25 degrees). In this experiment, all objects have the same shape and color making the task of following the same object for a long period of time very challenging. The plots demonstrate the success of our method in this very challenging scenario with object collisions and inter-object occlusions. For example, within a distance threshold of 20 pixels, which corresponds to the size of the smallest objects in the environment, our approach correctly follows more than 90% of objects during the rollout of 30 frames in all three considered camera viewpoints (top-view, 45 degrees and 25 degrees). **Please see also the supplementary videos "one_class*.mp4".**

Synthetic videos	5 fr.	10 fr.	30 fr.
Ours, top view	100	100	100
Ours, 45° tilt	99.3	96.2	96.2
Ours, 25° tilt	99.3	90.1	90.1
Linear motion baseline	81.1	67.8	59.7

Table 4. Percentage of predictions within a 20-pixel neighborhood around the target as a function of rollout length measured by the number of frames. 20 pixels corresponds to the size of the smallest objects in the dataset.

Model	Linear	MLP	Proba-RecIntNet (ours)
L2 dist. to target	28/71	19/43	**12/22**

Table 5. **Trajectory prediction on real videos.** Average Euclidean (L2) distance (in pixels in a 200 by 200 image) between predicted and ground truth positions, for a prediction horizon of 5 frames / 10 frames.

are recorded with a Microsoft Kinect2 device, including RGB and depth frames. The setup is similar to the one generated with Pybullet, recorded with a top camera view and containing 4 balls and a variable number of static blocks (from 0 to 3). Here again, the borders of the frame coincide with the walls of the box. Taking as input object segmentation of the first two frames, we use our model to predict object trajectories through the whole video (see Figure 4). We use the model trained on top-view synthetic Pybullet videos, without fine-tuning weights. We measure the error between predictions and ground truth positions along the roll-out. Results are shown in Table 5 and clearly demonstrate that out approach outperforms the linear and MLP baselines and makes reasonable predictions on real videos.

14. Future prediction (pixels): Comparison with baselines

We evaluate the error of the mask and depth prediction, measured by the training error described in detail in 8. Here, we compare our model to a CNN autoencoder (Riochet et al., 2018), which directly predicts future masks from current ones, without explicitly modelling dynamics of the individual objects in the scene. Note this baseline is similar to (Lerer et al., 2016). Results are shown in Table S1. As before, the existence of external occluders or the presence of tilt degrades the performance, but even in this case, our model remains much better than the CNN autoencoder of (Riochet et al., 2018).

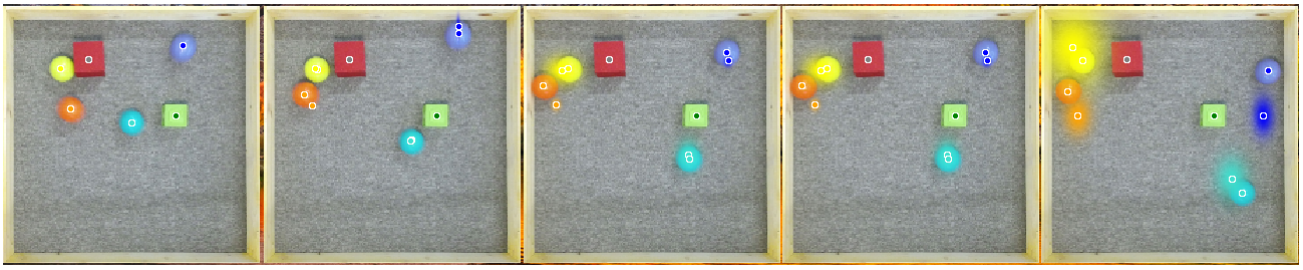

Figure 9. Example of prediction for a real video, with a prediction span of 10 frames. The small colored dots show the predicted positions of objects together with the estimated uncertainty shown by the colored "cloud". The same colored dot is also shown in the (ground truth) center of each object. The prediction is correct when the two dots coincide. (see additional videos).

	Top view	Top view+ occlusion	45° tilt	25° tilt	15° tilt
CNN autoencoder (Riochet et al., 2018)	0.0147	0.0451	0.0125	0.0124	0.0121
NoProba-RIN	0.0101	0.0342	0.0072	0.0070	0.0069
Proba-RIN	0.0100	0.0351	0.0069	0.0071	0.0065

Table 5. Aggregate pixel reconstruction error for mask and depth, for a prediction span of two frames. This error is the loss used for training (described in the supplementary material). It is a weighted combination of mask error (per-pixel classification error) and the depth error (mean squared error).