Unsupervised Learning of Intuitive Physics from Videos

Ronan Riochet

PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à Ecole Normale Supérieure

Unsupervised Learning of Intuitive Physics from Videos

Soutenue par

Ronan Riochet

Le 30 juin 2021

École doctorale n°386

Sciences Mathématiques de Paris Centre

Spécialité

Informatique - Sciences Cognitives

Composition du jury :

Jakob Verbeek Research Scientist, Facebook AI Research	*Rapporteur*
Justin Wood Associate Professor, Indiana University	*Rapporteur*
Camille Couprie Research Scientist, Facebook AI Research	*Examinateur*
Josef Sivic INRIA, Ecole Normale Supérieure	*Directeur de thèse*
Ivan Laptev INRIA, Ecole Normale Supérieure	*Directeur de thèse*
Emmanuel Dupoux INRIA, EHESS, Ecole Normale Supérieure	*Directeur de thèse*

1794
ENS | PSL

à ma famille,

I think Nature's imagination is so much greater than man's, she's never going to let us relax.

(Richard Feynman)

Contents

Remerciements

Je voudrais en premier lieu remercier Emmanuel Dupoux pour m'avoir donné l'opportunité de faire cette thèse. Emmanuel, tu es pour moi un modèle comme chercheur et comme directeur de thèse. Merci pour ta vision d'ensemble et la confiance que tu m'as accordée durant projet. Merci pour tous les moments où cela n'a pas marché et où tu m'as convaincu de continuer.

In english, I would like to thank my two co-supervisors Josef Sivic and Ivan Laptev, who accepted to get involved in the project when we knocked at Willow's door in 2017. Josef, working with you has been inspiring, both scientifically and in the way you interact with your students. Ivan, thank you for your constructive ideas and for always trying to make things better.

I would like to thank the members of my thesis jury, and first of all the two rapporteurs, Jakob Verbeek and Justin N. Wood. I have a huge consideration for their work, which is why I'm so honored that they accepted to review this thesis. Justin, I had the chance to attend a presentation you gave at Université Paris-Descartes, I felt impressed as you were presenting work I could not even imagine possible (e.g. dozens of chickens in their private virtual reality room). Jakob, I did not have the chance to meet you in person but I really appraciated getting to know you; thank you again for all your insighful comments on the manuscrit.

Bien sûr, merci à Camille Couprie d'avoir accepté de faire partie de ce jury. J'ai depuis le début de ma thèse été inspiré par tes travaux de recherche, merci de m'avoir permis de travailler à tes côtés (et de m'avoir initié aux voitures autonomes !).

Merci à tous ceux avec qui j'ai eu la chance de collaborer: Mario Ynocente Castro, Mathieu Bernard, Véronique Izard, Adam Lerer, Rob Fergus, Mohamed Elfeki et Natalia Neverova. Merci aussi à Victor, Louis et Malachi; et ceux qui ont participé au projet IntPhys: Erwan, Marianne et Valérian.

Je remercie bien sûr mes co-bureaux successifs, Mathieu, Benjamin, Haji, et bien

entendu l'étoile team Antoine, Ignacio et Hadrien.

Merci à mes collègues de l'équipe CoML/LSCP-model/Bootphon: Xuan-Gna d'abord sans qui nous ne serions rien, Catherine, Rachid, Neil, Rahma, Thomas, Mathieu, Elin et tous les autres. Un grand merci également à toute l'équipe Willow, Sierra et au quatrième étage en général: Yana, Thomas K., Loïc, Julia, Vijay, Adrien, Thomas E., Yann, Justin, Igor, Robin, Dimitri, Zongmian, Raphaël, Loucas, Grégoire, Jean-Paul et Jean.

Merci à Malo pour toutes les discussions, ses conseils et son aide; à Alexandre et Thomas pour leur soutien et leur patience. Le reste de l'équipe Milvue pour les bons moments passés ensembles: Aïssa, Oumaya, Hamza, Pierre, Ania, Clara, Liwa, Guillaume, Mohammed-Ali.

Merci aussi à mes amis de toujours, Antoine, Arthur, Capucine, Emeline, Hélène, Kevin, Léonard, Louise, Pierre (une deuxième fois), Rita, Seb, Sélim (et Hajar bien sûr, même si elle n'est pas dans le groupe whatsapp...), d'être présents depuis des années, et pour encore longtemps j'espère.

Evidemment, merci à mes parents Evelyne et Denis, ma tante Claudine, mes grands-parents et toute la famille pour leur soutien inconditionnel. Merci à mon frère Ludo, Léa, et leur petite Lison qui acquérait la physique intuitive quand j'écrivais ma thèse dessus. J'espère qu'elle lira ce manuscrit un jour !

Merci à Naëma pour son amour, sa patience et les moments passés ensembles.

Chapter 1

Introduction

Contents

1.1 Goals

Intuitive physics is often described as the untrained ability to understand the physical world; it has been commonly observed, and studied among infants, adults and animals. It allows one to catch a ball thrown in the air by anticipating its trajectory, or to build up a pile of dishes in the sink without casualties.

In the field of computer science however, modelling physics largely relies on rigorous mathematics equations which, while excelling at some complex tasks, fail when part of the system is unknown. For example, autopilots accurately land planes

using GPS coordinates and environment variables, but robots still struggle to stack piles of objects they see for the first time (Furrer et al. (2017)).

The goal of this thesis is to explore the ability for a system to learn intuitive physics from experience. Such a system would look at various videos of object interactions to learn the underlying physical regularities.

1.2 Motivations

Among applications that would benefit from such intuitive physics, there are:

Robotics. In robotics, consequent efforts have been made to automatically train a system to make a sequence of decisions. In this field, called *Reinforcement Learning*, model-based approaches consist of anticipating the outcome of different actions before choosing the best one. A model capable of predicting the physical consequences of its actions could achieve its task faster and safer.

Autonomous Driving. To move safely with little or no human input, autonomous vehicles shall anticipate possible obstacles arising in their surrounding environment. In many cases, this requires to understand physical interactions between objects.

Tracking. Video tracking consists of locating and linking moving objects in a video sequence. It has a variety of uses, some of which include: human-computer interaction, security, augmented reality, traffic control. Most tracking systems rely on two components: the *data model*, which identifies objects in the frames, and the *motion model*, which accounts for their dynamics. Because the motion model aims at discriminating possible and impossible trajectories, it would largely benefit from more robust priors on physics.

Cognitive Sciences. Besides applications in computer science, *reverse engineering* intuitive physics acquisition, i.e., building a system that mimics infant's achievements, could help understand the early infant development, and how physical intuitions arise in the human mind (see Dupoux (2018) for a similar approach in language acquisition).

1.3 Intuitive Physics of Objects in Infant Development

The process in which human beings acquire these concepts (and others, see Carey (2009)) has been debated for years. For empiricists, like John Locke, William James, or Jean Piaget, the initial state of infant cognition is limited to perceptual or sensory-motor representations:

- John Locke (1632-1704) thought of the mind as a "blank tablet" (*tabula rasa*) with sensory perceptions. He argued that ideas come from experience, and that no principle of reason is innate in the human mind (Locke (1689)).
- William James (1842-1910) famously believed that: *The baby, assailed by eyes, ears, nose, skin, and entrails at once, feels it all as one great blooming, buzzing confusion; and to the very end of life, our location of all things in one space is due to the fact that the original extents or bignesses of all the sensations which came to our notice at once, coalesced together into one and the same space. There is no other reason than this why "the hand I touch and see coincides spatially with the hand I immediately feel."* (James (1890))
- Jean Piaget (1896-1980) proposed that infants begin life with a repertoire of sensorimotor representations, achieving truly symbolic representations only at the end of second year of life (Piaget (1954))

In the end of the XX^{th} century, an alternative to this empiricist picture emerged with the work of psychologists like René Baillargeon, Randy Gallistel, Rochel Gelman, Alan Leslie, Elizabeth Spelke or Susan Carey. These writters shared the view that human cognition, like that of all animals, begins with highly structured innate mechanisms designed to build representations with specific content (Carey (2009)). In this thesis we investigate such structured mechanisms in artificial systems (see Chapter 2 for a review), in an attempt to understand those that are needed to build physical intuitions. In chapter 3, we draw inspiration from the works of these psychologists to build an evaluation procedure for intuitive physics in artificial intelligence systems.

1.3.1 Violation of Expectation Paradigm

Evaluating the early acquisition of intuitive physics concepts is difficult, as young infants have no access to language or advanced manipulation. Thus one cannot simply "ask" the infant about their understanding or assess their ability to perform complex tasks. For that reason, many infant development experiments rely on violation-of-expectation tasks: given a physical rule, infants are shown normal events (often referred to as *possible*) *versus* events breaking the physical rule in question (often referred to as *impossible*). After an habituation phase where the infant is shown several normal events, we measure their attention time in front of possible and impossible events, which is interpreted as the *surprise* expressed by the infant. The hypothesis is that infants' attention time is longer when being shown impossible events than possible events. If this hypothesis is statistically true, we (abusively) say the infants "understand" this physical rule.

In Chapter 3, we design such a procedure to evaluate intuitive physics in systems, on three physical rules: *Object Permanence*, *Shape Consistancy* and *Spatio-Temporal Continuity*.

1.3.2 Core Knowledge of Objects

In this section we describe a subpart of the *Core Cognition* described in Carey (2009), which deals with the notion of *Object*. Even though we describe it for human beings, this core notion of object has been shown to be shared by other animals (Gallistel (1990)).

Perception of objects. Young infants have expectations that objects are bounded and cohesive over time. Cheries et al. (2008) experiment with a crawling task on 10-month-old infants to demonstrate that they fail to track objects when broken into two pieces, suggesting that violations of cohesion disrupt infants' object tracking abilities. In another experiment, Needham (1999) show that 4-month-old infants are more likely to use shape rather than color and pattern differences to find object boundaries (it seems to remain the case until around 11 months).

Object permanence. Object permanence is the concept that objects continue to exist when they are occluded. Baillargeon et al. (1985) introduced a method devised to test object permanence in young infants based on the violation-of-expectation procedure described above. In their experiment, five-month-old infants are habituated to a screen that moves back and forth through a 180-degree arc (see figure 1.1). After this habituation phase, a box is centered behind the screen and infants are shown possible and impossible events. In the possible event, the screen stops when it reaches the occluded box; in the impossible event, the screen moves through the space occupied by the box. Results indicated that infants look reliably longer the impossible event, thus authors drew the following conclusion:

> *Contrary to Piaget (1954) claims, infants as young as 5 months of age understand that objects continue to exist when occluded. The results also indicate that 5-month-old infants realize that solid objects do not move through the space occupied by other solid objects. (...)*

This experiment was reconducted in Baillargeon (1987) with 3.5 to 4.5-month-old infants, showing that the 4.5-month-olds, and a portion of the 3.5-month-olds infants looked reliably longer at the impossible than at the possible event. Aguiar and Baillargeon (1999) also investigated 2.5-month-old infants' reasoning about occlusion events. They focused on infants' ability to predict whether an object remains hidden or becomes temporarily visible when passing behind an occluder with an opening in it. In Chapter 3, we draw inspiration from these experiments to create one of the blocks of our IntPhys Benchmark.

Continuity & solidity Spelke et al. (1992) provided evidences for early-developing capacities of young infants to reason about object motion. They showed that infants as young as 2.5 to 3-month-old already exhibit two physical conceptions: *continuity* and *solidity*. The term "continuity" refers to the fact that objects move only on connected paths and do not jump from one place and time to another; "solidity" referring to the fact that objects move only on unobstructed paths and no parts of two distinct objects coincide in space and time.

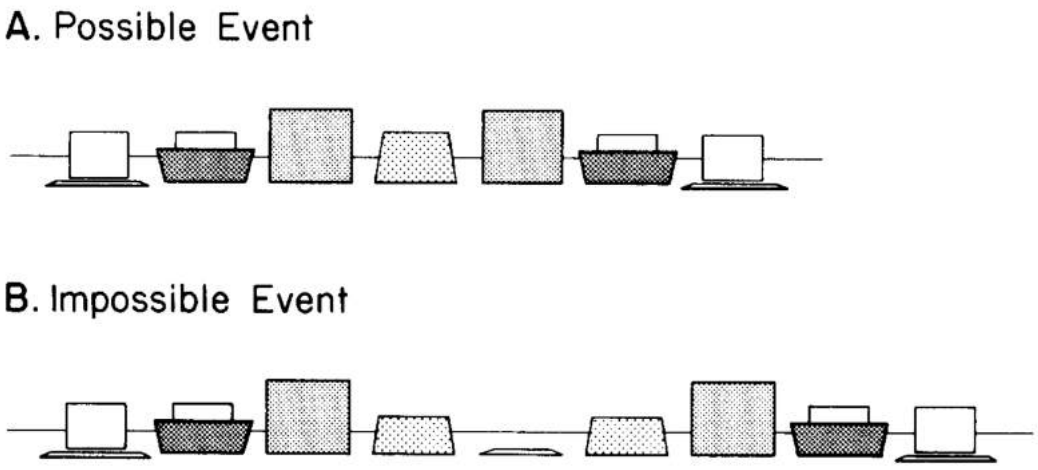


Figure 1.1: Schematic representation of the possible and impossible test events in the object permanence experiment, from Baillargeon et al. (1985).

Inertia & gravity In addition to their experiments on continuity and solidity, Spelke et al. (1992) demonstrated that such young infants (3-month-old) fail at expressing intuitions about *inertia*; the fact that objects do not change their motion abruptly and spontaneously, and *gravity*; that objects move downward in the absence of support. Other experiments showed that these notions arise later in the development: at around 7 months for gravity (Kim and Spelke (1992)) and from 8 to 10 months for intertia (Spelke et al. (1994)).

Conservation of properties. Infants don't expect objects' intrinsic properties like size, shape, pattern, or color, to change with no reason. Wilcox (1999) demonstrated that 4.5-month-olds use both shape and size to discriminate objects during occlusion events. It is around 7.5-months-old that they use pattern, and only at 11.5 months that they use color to reason about object identity.

Figure 1.2 shows an overview of the acquisition of intuitive physics in infant development. An exhaustive review can be found in Hespos and Vanmarle (2012).

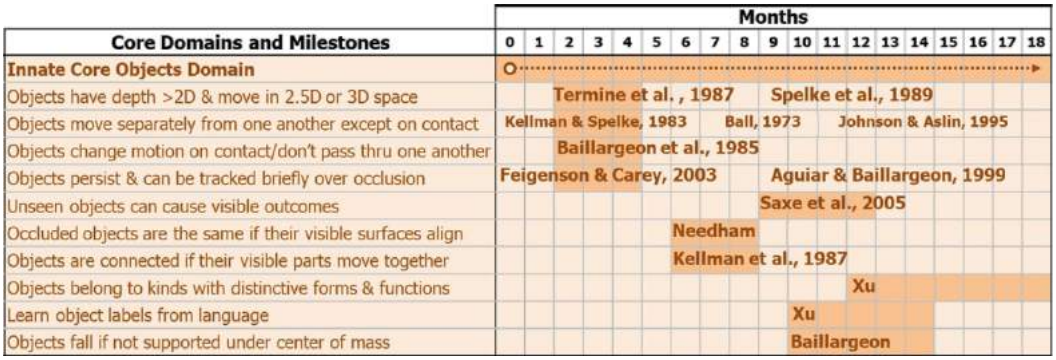


Figure 1.2: Approximate timeline of intuitive physics in infant development (from DARPA *Machine Common Sense* project presentation).

1.3.3 A Bootstrapping Problem

Bottom-up & Top-down Mechanisms

In psychology, we define as *bottom-up* processes those that arise from sensory reception and do not require any knowledge or prior on the world. For example, in the context of intuitive physics, infants' visual system bounds, locates and identifies objects, resulting in a sequence of object proposals over time. This process is said to be bottom-up, as it arises from the visual system with no prior on the underlying physics.

On the other hand, we define as *top-down*, processes that are influenced by our knowledge, or prior, about the world. Violation of expection experiments show evidence of these processes: infants express surprise when faced events that contradict their knowledge of the physical world.

From a learning point of view, these bottom-up (Needham (1999); Cheries et al. (2008); Wilcox (1999)) and top-down mechanisms (Baillargeon et al. (1985); Baillargeon (1987); Spelke et al. (1992)) often appear to be interdependant. For instance in the object permanence experiment, infants are shown objects that disappear behind an occluder, then reappear. While they are visible, the visual system catches object cues; but because of changes in illumination or orientation, these object cues are noisy and tracking their identity is already non-trivial: we must understand that objects' position and appearance change smoothly in time. It becomes even harder

when an object gets occluded: we assume it continues to exist even if we don't see it, and predict its trajectory to anticipate when it will get out of occlusion. How can these intertwined processes mature together during the first months of life? This computational challenge is often called *bootstrapping problem*.

Bootstrapping Problem

In psychology, a "bootstrapping" process is a process in which *a system uses its initial resources to develop more powerful and complex processing routines, which are then used in the same fashion, and so on cumulatively* [1]. In language acquisition for example, the term is used to express the complexity of learning the rules for natural languages, given the few observable data and numerous ambiguities the child is being faced (Pinker (1987)).

The developmental psychologist Susan E. Carey proposed the term "Quinian bootstrapping", after the philosopher and logician Willard Van Orman Quine (1908–2000), to describe the theory that humans build complex concepts (including intuitive physics) out of primitive ones through a bootstrapping process.

Minimum viable physics for autonomous systems

In this introduction, we have presented works from psychologists sharing a similar view of human cognition: it begins with highly structured innate mechanisms, from which it builds specific representations through bootstrapping processes. In the next chapters we investigate what properties are required for a system to build such rich representations about the physical world. In particular, we explore different computational systems and compare their performances on tasks inspired from the infant development studies presented in this introduction.

[1] American Psychological Association, `dictionary.apa.org`

1.4 Contributions

In chapter 3, we create a consistent series of tests to evaluate intuitive physics in systems (Riochet et al. (2021)). Relying on the Violation Expectation Paradigm described above, we designed a benchmark based on three Intuitive Physics concept: *object permanence*, *shape consistency* and *trajectory continuity*. We also conducted human studies and compared results with two pixel-based baseline models. To our best knowledge, this work was the first to use Violation Expectation Paradigm to evaluate Intuitive Physics in systems and was followed by two other works: Piloto et al. (2018); Smith et al. (2019). This work is published in *IEEE Transactions on Pattern Analysis and Machine Intelligence* and has also been used by the DARPA for its *Machine Common Sense* project, to evaluate works on Intuitive Physics. We counted more than 20 teams evaluating their models on the benchmark in the last 3 years.

Our experiments from Riochet et al. (2021) show that CNN encoder-decoder structure (either trained in an adversarial procedure or not) are not enough to learn the type of physical regularities we considered, especially in the case of occlusions. In chapter 4 we design an object-based model, gifting the system with a notion of *objects*, interacting together and in which physics is compositional Riochet et al. (2020a). Our experiments on simulated videos showed we are able to perform object tracking and forward modelling, even when there were frequent occlusions. One could compare this structure imposed in the model as the Core Knowledge of Objects described in 1.3.2, in opposition to the pure empiricist hypothesis of a more general model learning the notion of object from visual inputs only.

Finally, in chapter 5, we adapt this approach to the case of a moving camera, applying it to two city driving datasets: one synthetic recorded with Carla (Dosovitskiy et al. (2017)), and the Cityscapes Dataset (Cordts et al. (2016)) made of real video sequences recorded in streets from 50 cities. In addition, we proposed a method to decouple ego-motion from objects' motion, making it easier to learn long term object dynamics.

Articles

– R. Riochet, M. Ynocente Castro, M. Bernard, A. Lerer, R. Fergus, V. Izard, and E. Dupoux. IntPhys: A Framework and Benchmark for Visual Intuitive Physics Reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021

– Ronan Riochet, Josef Sivic, Ivan Laptev, and Emmanuel Dupoux. Occlusion resistant learning of intuitive physics from videos. 2020b

– Ronan Riochet, Mohamed Elfeki, Natalia Neverova, Emmanuel Dupoux, and Camille Couprie. Multi-Representational Future Forecasting. November 2020a

Patents

Adam Kal Lerer, Robert D. Fergus, and Ronan Alexandre Riochet. *Differentiating Physical and Non-Physical Events*. Google Patents, October 2019

Oral presentation

– *Intphys: A Benchmark for Visual Intuitive Physics Understanding*. Workshop on Intuitive Physics, NIPS2016, Barcelona, Spain, 22 - December 2016.

– *Learning intuitive physics from videos*. Journées interdisciplinaires de l'Ecole Normale Supérieure, Paris, 22 - March 2019.

Open source projects

– Models presented in chapter 3: `github.com/rronan/IntPhys-Baselines`

– Unreal Engine environment for data generation presented in chapter 3 (partial contribution): `github.com/bootphon/intphys`

– Models presented in chapter 4: `github.com/rronan/Recurrent-Interaction-Network`

Chapter 2

Litterature Review: Intuitive Physics in Computer Vision and Artificial Intelligence

Contents

For centuries, physicists have been describing the world through mathematical equations that matched observations. With this formalism and the emergence of computer science came the possibility to simulate almost any kind of physical system, sometimes with an extreme precision. This allowed human kind to send satellites in orbit, cross an ocean in six hours or forecast weather for several days.

These programs are often tailor made for specific problems: involving mathematical tools that differs from one situation to another (e.g. rigid or soft body physics, fluid dynamics, etc.). With the rise of deep learning, researchers have tried to build systems that could discover these regularities from observations. In that scenario, instead of writting down a sequence of instructions, the researcher designs a model capable of learning those instructions from the observed data, through a so-called *training* phase. This way, Xingjian et al. (2015) predicted precipitation nowcasting with a deep learning architecture that was later used for biological age estimation (Rahman and Adjeroh (2019)), traffic flow prediction (Liu et al. (2017)) or video salient object detection (Song et al. (2018)).

While these machine learning approaches are still far less accurate than the traditional ones, showing ability to *learn* these rules from observations echoes with the mechanisms described in Chapter 1. In this thesis we restrict to the types of physical interactions described in the litterature of infant development, sometimes called *intuitive physics*. This chapter contains a review of the litterature on Intuitive Physics in Computer Vision and Artificial Intelligence.

2.1 Intuitive Physics in Controlled Environments

Modelling intuitive physics is inherently tied to our representation of the physical scene. Such representation may be RGB pictures, point clouds or more structured like object representations. Some of them, like the first one, have the advantage of being applicable to various real-life scenarios as videos are easy and cheap to record. In this section, we present works that were proposed to learn intuitive physics from controlled environments where structured object representation is available.

In such environment, objects are represented by their coordinate vector, either in 2D Chang et al. (2016); Battaglia et al. (2016) or 3D Mrowca et al. (2018); Li et al.

(2018b). In Battaglia et al. (2016), authors introduce the *Interaction Network*, a neural network taking as input objects' physical state at a given time and predicting their trajectory in the near future. To do so, they build a graph where each node is an object (described by its position, velocity and mass) and each edge describes interaction between these objects. By factorizing one model to predict these interactions, this approach is compositional and allows to take a variable number of objects as input. This is also the case for Chang et al. (2016) where, in addition, authors prune the graph from interactions involving objects that are too distant in space (see examples of scenes they consider in Figure 2.1). While these two works involve rigid objects in the 2D plane, Mrowca et al. (2018); Li et al. (2018b) use the same idea to model soft bodies and fluids in the 3D space. In that case, one body is itself described with several atomic parts (see Figure 2.2) which interact together, causing its deformation.

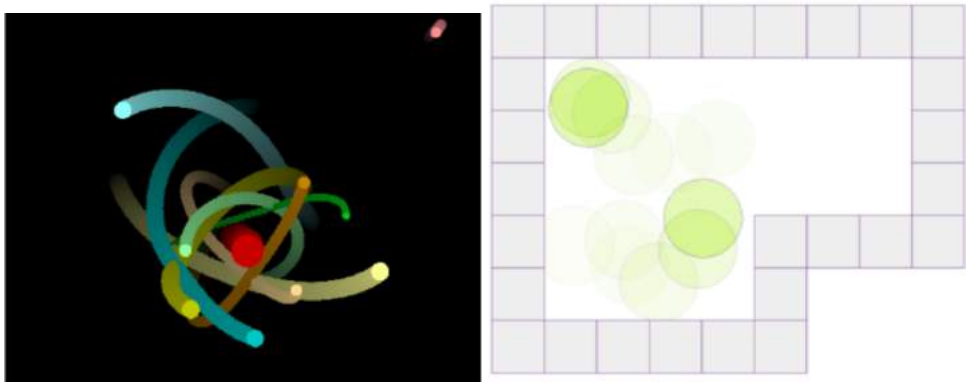


Figure 2.1: Example of physical scenes modelled in Battaglia et al. (2016) (left) and Chang et al. (2016) (right).

Battaglia et al. (2013) have also investigated a probabilistic model, which they call *Intuitive Physics Engine*. It uses Open Dynamics Engine (`www.ode.org`) as a rigid body simulator, running on an object-based representation of a 3D scene. This physics engine runs on multiple independent draws sampled from a probability distribution accounting for the observer's belief on some physical quantities (mass of object, precise location of a partially occluded object, etc.). This model shows similar behavior with humans on five distinct psychophysical tasks.

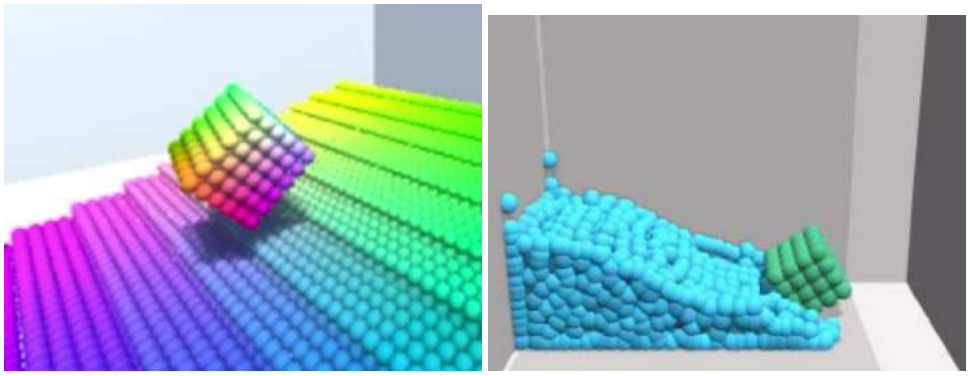


Figure 2.2: Example of physical scenes modelled in Mrowca et al. (2018) (left) and Li et al. (2018b) (right).

2.2 End-to-End Forward Prediction in Videos

2.2.1 Next-frame prediction

Other works have focused on predicting future frames in videos Mathieu et al. (2015); Pătrăucean et al. (2016); Wichers et al. (2018). One motivation for this task was that training a model for forward prediction would require the model to understand objects motion, thus both learn to represent objects and infer their dynamics. Ideally, this would be a way to learn visual features for object detection, without the need for expensive, human annotated, datasets. In practice however, these approaches have not yielded any consistent improvements in object detection, compared to fully-supervised approaches described in Section 2.3.1.

2.2.2 Using additional information

Other works have proposed using more supervision in the task of future frame prediction. Luc et al. (2017, 2018); Couprie et al. (2018) predict future segmentation, allowing to focus on object motion without having to predict textures and changes in lighting.

Keypoints have also been used in forward prediction. Villegas et al. (2018) encode

objects in a video as a time series of keypoints, then use a Long Short-Term Memory (LSTM) network Hochreiter and Schmidhuber (1997) to predict their future pose. Finally, they train an image generator to predict the future frame from the initial frame and the predicted pose.

Finally, optical flow, the instantaneous velocity of pixels moving in a video; has been used as a cue to infer objects' velocities. Liang et al. (2017) propose a generative adversarial network (GAN) to predict both future frame and optical flow.

2.3 Intuitive Physics of Objects in Videos

In this section we first describe the notion of object in computer vision, then present works on learning intuitive physics of objects from visual inputs.

2.3.1 Notion of Object in Computer Vision

In computer vision, the notion of object itself varies, along with methods used to detect them. In most cases, an object detection is defined by a bounding box: a rectangle around the object in the picture, and a label specifying the kind of object that is detected. It can be completed with an instance mask which tells, for each pixel, if it is part of the object or not. Additional information, like keypoints, can extend this detection. Examples of such detections can be found in Figure 2.3.

Supervised object detection. Supervised object detectors are models trained on large datasets of images paired with the list of visible objects, along with their localization. Girshick et al. (2014); Girshick (2015); Ren et al. (2017) propose a *Region-based Convolutional Network* (RCNN), that use a pre-trained convolutional neural network to bottom-up region proposals, in order to localize and segment objects. Redmon et al. (2016) propose a model that performs detection at a rate faster than 24 images, making it suitable for real time application on videos streams.

Lin et al. (2017) propose a *Feature Pyramid Network* (FPN) which efficiently computes pyramid representation. This rich representation can be used in a Faster-RCNN system (Ren et al. (2017)) to improve performance with marginal extra cost.

Finally Tan et al. (2020) propose key optimizations to improve efficiency of object detectors, resulting in the current state-of-the-art model.

The two main datasets, Pascal VOC Everingham et al. (2010) and Microsoft COCO Lin et al. (2014) propose a large number of images with annotated object instances, as well as a test set and evaluation benchmark. Microsoft COCO contains 300 000 fully segmented images, where each image has an average of 7 object instances from 80 categories. Pascal VOC, on its side, contains only 20 categories. It is also common to pretrain some parts of the model on large image classification datasets (e.g., ImageNet Russakovsky et al. (2015)) to improve extracted visual features.

Figure 2.3: Example of object detections with bounding boxes and labels.

Figure 2.3: Example of object detections with keypoints.

Figure 2.3: Example of object detections with instance masks and labels, from `www.github.com/facebookresearch/detectron2`.

Unseen/Salient object detection. While supervised models offer state of the art results on detection and segmentation benchmarks (Everingham et al. (2010); Lin et al. (2014)), they fail to detect objects that are too different from those of the training set. Other approaches propose to detect and segment objects of any shape by focusing on cues like contours, saliency, depth estimation.

Visual saliency detection aims to discover regions in an image that look the most like a object. We can distinguish *bottom-up* and *top-down* approaches. In bottom-up approach (like Tu et al. (2016)), low-level visual features (e.g. edges, texture) play a central role, regardless of the semantic content. In contrast, top-down approaches, like Yang and Yang (2017), use priors about object categories and spatial context to make their prediction. In practice, these two mechanisms combine well: the former proposes object region candidates while the latter prune these candidates with respect to a prior knowledge about objects. A litterature review on object detection (in date of 2019) can be found in Zhao et al. (2019)

Detecting objects from videos. Methods presented above detect objects from still images, but videos offer additional information which should help making better predictions (e.g., distinguishing ambiguities due to occlusions and/or lighting conditions). Zhu et al. (2017) and Li et al. (2018a) investigate flow guided, end-to-end methods for video object detection. Zhu et al. (2017) propose to aggregate features from a reference frame with thoses of nearby frames, based on optical flow information. These aggregated features are used to predict more robust object detection in the reference frame. Li et al. (2018a) propose a similar idea, encoding sequential feature evolution with LSTM networks. See also Agarwal et al. (2016) for a review of on optical flow litterature prior to 2016.

Other approaches. Finally, other works have used proxy tasks, like video colorization, to detect and track objects (Vondrick et al. (2018)). Greff et al. (2019) and Burgess et al. (2019) propose methods to learn - without supervision - to segment images into interpretable objects with disentangled representations. In particular, they train variational autoencoders (VAE) to reconstruct input images, given the prior that they contain several objects. While this method works well on synthetic datasets like Johnson et al. (2016), they fail to generalize on real images like those of ImageNet (Russakovsky et al. (2015)).

2.3.2 Learning the Dynamics of Objects in Videos

Several works have attempted to learn physical regularities from videos. Compared to coordinate trajectories, working from a sequence of video frames is challenging, because:

- High dimension: There are more 10×10 distincts images than the total number of images seen by all humans through out history [1].
- Occlusions and changes in illuminations make information noisy or missing.
- The losses used on pixels only partially reflects the layout of the physical scene (see Figure 2.4).

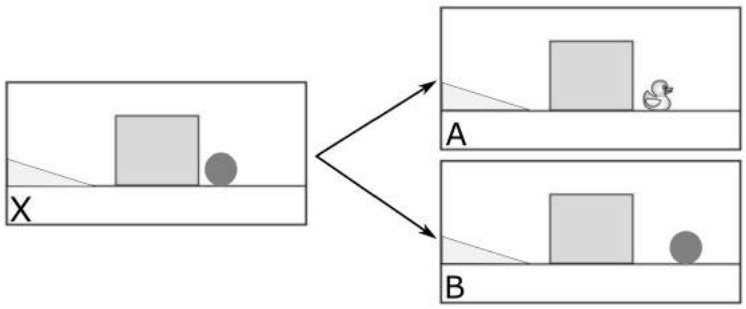


Figure 2.4: Is image X closer to image A or B? Even though they show two different objects, the L2 distance between A and X is lower than the one between B and X

Simple and planar scenes. Some works focus on simple images, with low resolution (usually around 100x100 pixels or lower), with very simple shapes and no changes in illumination. Fraccaro et al. (2017) investigate a Kalman filter based variational auto-encoder, that simultaneously learns two disentangled representations

[1] Number of 10×10 distincts images $256^{10 \cdot 10} = 6.7 \cdot 10^{240}$ *vs* images seen by 50 billion people, during 20 billion seconds, with 30 images per second: $3 \cdot 10^{21}$. From `http://helper.ipam.ucla.edu/publications/gss2011/gss2011_9841.pdf`

for videos: one accounting for object recognition and the other for their dynamics. Experiments are done on 32×32 binary pixels videos, with only one object.

Watters et al. (2017) and van Steenkiste et al. (2018) learn the dynamics of several objects in 64×64 pixels videos. Watters et al. (2017) use a visual encoder to estimate the state of every object, then apply an Interaction Network (Battaglia et al. (2016)) to predict future states. van Steenkiste et al. (2018) explore the use of a *Relational Neural Expectation Maximization*, a Neural Expectation Maximization (Greff et al. (2017)) endowed with a relational mechanism also similar to Battaglia et al. (2016).

Although input videos are a lot simpler than those from real life applications, these works have the advantage of exploring end-to-end approaches, which do not need to be pretrained with annotated data.

Controlled or synthetic realistic scenes. Going forward to learning intuitive physics from real-life videos, many works have tried to simulate simple scenes in 3D virtual environments. They either use game engines like UnrealEngine (Epic Games (2019)) or Unity (Technologies (2005)), or more research oriented libraries like PyBullet (Coumans and Bai (2016)) or MuJoCo (Todorov et al. (2012)).

Lerer et al. (2016); Li et al. (2016); Mirza et al. (2017); Groth et al. (2018); Zhang et al. (2016) have focused on predicting the stability of piles of blocks from still images (see examples in Figure 2.5). They use CNN based image classifiers taking as input an image of a block tower and returning a probability for the tower to fall. Lerer et al. (2016); Mirza et al. (2017) also include a decoding module to predict final positions of these blocks. Groth et al. (2018) investigate the ability of such a model to actively position shapes in stable tower configurations.

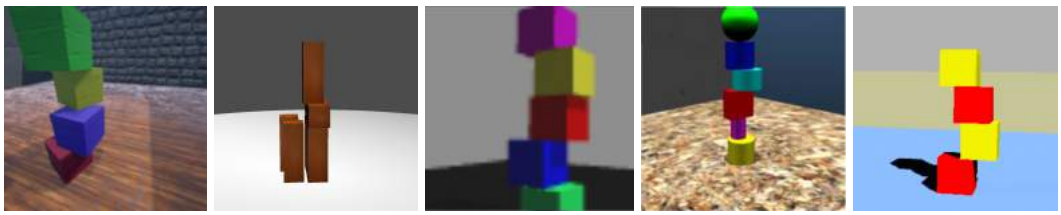


Figure 2.5: Examples of input images from Lerer et al. (2016); Li et al. (2016); Mirza et al. (2017); Groth et al. (2018); Zhang et al. (2016) (from left to right).

Other works have investigated more complex physical scenes in synthetic videos. Wu et al. (2017a) rely on a shallow object detector estimating position and physical states of simple shapes. The resulting object trajectories are used by a neural physics engine like in Chang et al. (2016).

Similarly to Chapter 3, Smith et al. (2019) and Janner et al. (2019) proposed Violation-of-Expectation based datasets to evaluate intuitive physics understanding. They also investigate models including modules for parsing the scene as objects and predicting their future motion.

Real videos. Finally, we review works that have focused on intuitive physics and forward modelling in real videos. Srivastava et al. (2015) use Long Short Term Memory (LSTM) networks to learn representations of video sequences, also predicting the future sequence. Pătrăucean et al. (2016); Lotter et al. (2017); Sun et al. (2019) propose models based on Convolutional Long Short-Term Memory cells (ConvLSTM), similar to Xingjian et al. (2015), for video forcasting. Rather than focusing on future frame, Sun et al. (2019) predict future instance segmentation. This approach allows to focus more on object position and orientation rather than changes in texture or luminosity, and is also the direction we chose in Chapter 5. Mathieu et al. (2015) investigate different losses for preserving objects' sharpness in video forecasting. Ranzato et al. (2016) have proposed a baseline inspired by language models to predict future frames in videos. Vondrick et al. (2016) propose a generative adversarial network with a spatio-temporal convolutional architecture

that disentangles the foreground and background.

Finn et al. (2016) use a LSTM-based approach for a model-based reinforcement learning task, for robotic manipulation. Xu et al. (2019b) propose a model learning physical object properties from dynamics interaction (through a robot's arm) and the resulting visual observations.

2.3.3 Inferrence of physical properties

Wu et al. (2016) construct a dataset, called *Physics101*, with videos of various objects interacting one with each other. Taking inspiration from Carey (2009), they design four different scenarios:

- **Ramp**: Objects are put on an inclined surface and may either slide down or stay static, due to gravity and friction.
- **Spring**: Objects are hung to a spring, gravity on the object stretching the spring.
- **Fall**: Objects are dropped in the air and freely fall onto various surfaces.
- **Liquid**: Objects are dropped into some liquid and may float or sink at various speeds.

Like in Battaglia et al. (2013), authors use a hard-coded *physical world simulator*, predicting object dynamics given their physical properties: mass, volume, friction coefficient, restitution coefficient, elasticity. They train a neural network to estimate these quantities from observations, given the constraints encoded in the physical world simulator.

Summary

We have presented different works on intuitive physics as explored in the artificial intelligence and computer vision litteratures. Although these works agree on the problems they tackle, the models considered as well as the data they evaluate on differ a lot. In this section, we intend to summarize those differences in light of motivations presented in Chapter 1.

The first axis of variation we consider is the type of data that is used for the experiments. It goes from trajectories simulated in the cartesian space to real videos of complex scenes. The main categories are:

(i) Perfect object trajectories in the cartesian plane (2D). Each object is represented by its x-y coordinates and physical intrinsic properties. Typically, such trajectories come from simulated worlds as shown in Figure 2.1.

(ii) Perfect object trajectories in the cartesian space (3D): similar to the previous one, but in 3D (see Figure 2.2). Works like Battaglia et al. (2016); Chang et al. (2016); Mrowca et al. (2018); Li et al. (2018b) show that neural networks can learn such trajectories and make long term predictions. However, this is to be put in perspective with the fact that traditional physics engines (included those used to simulate these data!) do perform extremely good on these tasks.

(iii) Simple 2D videos: videos with simple objects on a 2D plane, with no occlusion or changes in illumination. This includes videos of billiard boards with distinctly colored balls, as well as synthetic videos created in 2D environments.

(iv) Controlled 3D synthetic videos: recorded in 3D virtual environments with fixed camera, few objects or changes in background and illumination.

(v) 3D synthetic videos: more complex videos with camera in motion and high diversity in objects categories and motion. This includes city-driving datasets like Dosovitskiy et al. (2017).

(vi) Controlled real videos: recorded with a standard camera, but in an controlled environment. The camera is fixed and the scene involves only few objects with simple appearance. These scenes are almost perfectly segmented by standard object detectors, and simple tracking heuristics successfully compute object trajectories.

(vii) Unconstrained real videos from everyday life. Objects can be of diverse appearance and their motion may induce occlusion, causing standard object detectors to fail at segmenting some objects in some frames. Simple tracking heuristics may fail.

The second axis we consider is the notion of physics, or the prior on physics, that is *given* to the model. This reflects the choices that are made in the model's design, such as the structure of the neural network, to allow it modelling the physical world. We propose the following partition:

(a) No prior on physics: a general model capable of learning regularities from a training dataset (either videos or trajectories), with no specific design aimed to learn intuitive physics or object dynamics.

(b) Physics is shared per object (but learnt): the model is still general, but is applied to each object individually. To work on videos, this requires the use of a visual encoder (either trained end-to-end or including a pre-trained object detector).

(c) (ii) + Physics is shared per pairwise interaction: factorizing one model to learn all pairwise interactions.

(d) (iii) + Movement is 2nd order (continuity of trajectories): position is the derivative of velocity, which is the derivative of acceleration. Applying constraints on acceleration or velocity helps to smooth trajectories, especially when observations are noisy or partially missing. A notable version, the *Kalman filter*, can be used in the tracking of objects in videos.

(e) Traditional physics engine inside. The system includes a traditional physics engine to make forward predictions, sometimes using sampling to account for uncertainty in observations.

In Figure 2.6, we classify related works presented above in regards to these two axes of variation.

2.4 Beyond Object Tracking: Event Decoding

Multiple Object Tracking (MOT) consists in locating multiple objects, maintaining their identities, and yielding their individual trajectories given an input video. Such "objects" can be pedestrians Yang et al. (2011); Pellegrini et al. (2009), vehicles

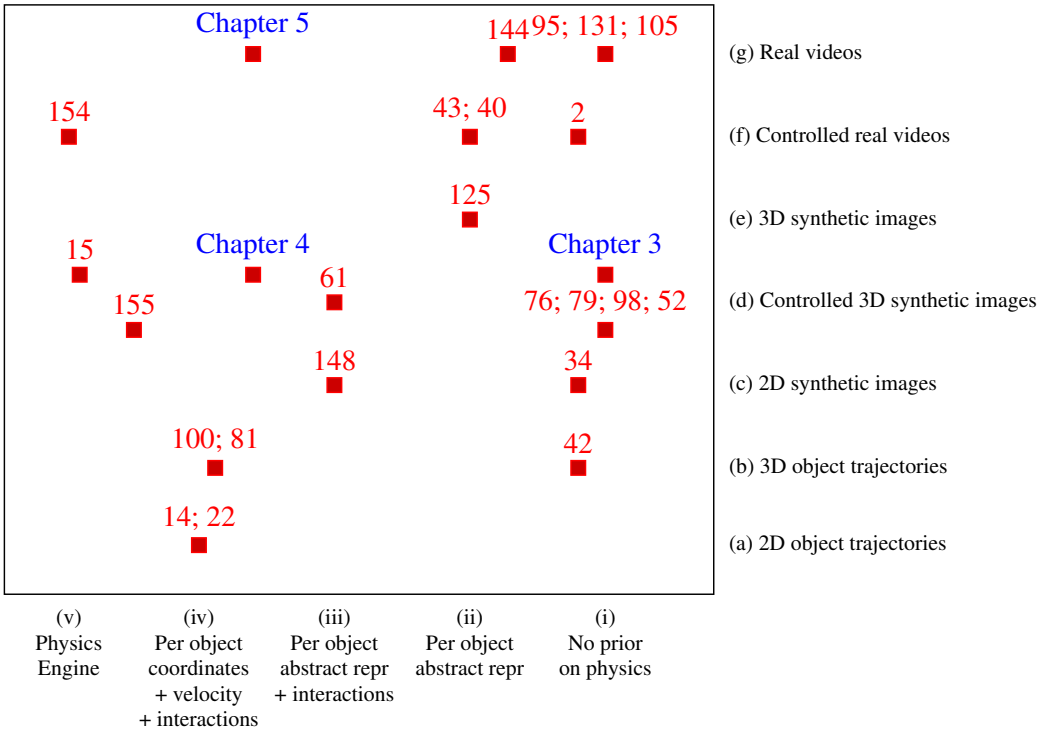


Figure 2.6: Scatter plot of works on intuitive physics in computer vision and artificial intelligence.

Koller et al. (1994); Betke et al. (2000), sport players Lu et al. (2013), animals Spampinato et al. (2008), etc. An exhaustive review can be found in Luo et al. (2017).

We call *event decoding* the problem of assigning to a sequence of video frames $F = f_{t=1..T}$ a sequence of underlying object states (i.e., object positions, velocities, appearance, mass, etc.) $S = s_{t=1..T}^{i=1..N}$ that can explain this sequence of frames. Within a generative probabilistic model, we therefore try to find the state $\hat{S}$ such that:

$$\hat{S} = \underset{S}{\operatorname{argmax}}\, P(S|F, \theta) \tag{2.1}$$

where θ is a parameter of the model.

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 Neufeld (1999).

In practice, this optimization problem 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 chapter 4 we will attempt to make this problem tractable. In first place, we use off-the shelf instance mask detectors presented above to operate in mask space and not in pixel space. Second, we approximate these probabilistic model with two Neural Networks, one rendering model and one physical model. We will design them as compositional, which may be defined as:

- *"Compositionality is an embodiment of faith that the world is knowable, that one can tease things apart, comprehend them, and mentally recompose them at will"* (Alan Yuille)

- *"The world is compositional or God exists"* (Stuart Geman)

Chapter 3

IntPhys: A Benchmark for Visual Intuitive Physics Understanding

Abstract

In order to reach human performance on complex visual tasks, artificial systems need to incorporate a significant amount of understanding of the world in terms of macroscopic objects, movements, forces, etc. Inspired by work on intuitive physics in infants, we propose an evaluation benchmark which diagnoses how much a given system understands about physics by testing whether it can tell apart well matched videos of possible versus impossible events constructed with a game engine. The test requires systems to compute a physical plausibility score over an entire video. It is free of bias and can test a range of basic physical reasoning concepts. We then describe two Deep Neural Networks systems aimed at learning intuitive physics in an unsupervised way, using only physically possible videos. The systems are trained with a future semantic mask prediction objective and tested on the possible versus impossible discrimination task. The analysis of their results compared to human data gives novel insights in the potentials and limitations of next frame prediction architectures.

This work was led in collaboration with Mario Ynocente Castro, Mathieu Bernard, Adam Lerer, Rob Fergus, Véronique Izard and Emmanuel Dupoux. The preprint is currently under review for TPAMI.

Contents

3.1 Introduction

Despite impressive progress in machine vision on many tasks (face recognition Wright et al. (2009), object recognition Krizhevsky et al. (2012); He et al. (2016), object segmentation Pinheiro et al. (2015), etc.), artificial systems are still far from human performance when it comes to common sense reasoning about objects in the world or understanding of complex visual scenes. Indeed, even very young children have the ability to represent macroscopic objects and track their interactions

through time and space. Just a few days after birth, infants can parse their visual inputs into solid objects Valenza et al. (2006). At 2-4 months, they understand object permanence, and recognize that objects should follow spatio-temporally continuous trajectories Kellman and Spelke (1983); Spelke et al. (1995). At 6 months, they understand the notion of stability, support and causality Saxe and Carey (2006); Baillargeon et al. (1992); Baillargeon and Hanko-Summers (1990). Between 8 and 10 months, they grasp the notions of gravity, inertia, and conservation of momentum in collision; between 10 and 12 months, shape constancy Xu and Carey (1996), and so on. Reverse engineering the capacity to autonomously learn and exploit intuitive physical knowledge would help building more robust and adaptable real life applications (self-driving cars, workplace or household robots).

Although very diverse vision tasks could benefit from some understanding of the physical world (see Figure 3.1), modeling of intuitive physics has been mostly developed through some form of future prediction task Battaglia et al. (2016),Chang et al. (2016),Xue et al. (2016),Fraccaro et al. (2017) and reinforcement learning Veerapaneni et al. (2020). Being presented with inputs that can be pictures, video clips or actions to be performed in the case of a robot, the task is to predict future states of these input variables. Future prediction objectives have a lot of appeal because there is no need for human annotations, and abundant data can be collected easily. The flip side is that it is difficult to find the right metric to evaluate these systems. Even though pixel-wise prediction error can be a good loss function, it is not particularly interpretable, depends on the scale and resolution of the sensors making cross datasets comparison difficult, may not even rank the systems in a useful way: a good physics model could predict well the position of objects, but fail to reconstruct the color or texture of objects. In addition, even though the laws of macroscopic physics are deterministic, in practice many outcomes are stochastic (this is why people play dice). In other words, the outcome of any interaction between object is a distribution of object positions, making the evaluation problem even harder.

Here, we propose to use an evaluation method which escapes these problems by using the prediction error not directly as a metric, but indirectly as informing a forced choice between two categories of events: *possible* versus *impossible events*. The intuition is the following. If a model has learned the laws of physics, it should be able to predict relatively accurately the future in video clips that show possible

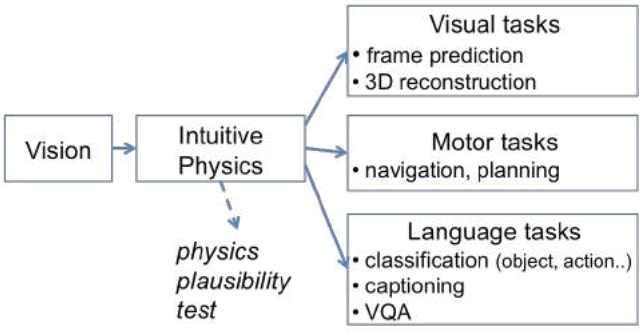


Figure 3.1: Popular end-to-end applications involving scene understanding and proposed evaluation method based on physical plausibility judgments. 'Visual' tasks aim at recovering high level structure from low level (pixel) information: for instance, recovering 3D structure from static or dynamic images (e.g., Chang et al. (2015); Choy et al. (2016)) or tracking objects (e.g., Kristan et al. (2016); Bertinetto et al. (2016)). 'Motor' tasks aim at predicting the visual outcome of particular actions (e.g., Finn et al. (2016)) or to plan an action in order to reach a given outcome (e.g. Oh et al. (2015)). 'Language tasks' requires the artificial system to translate input pixels into a verbal description, either through captioning Farhadi et al. (2010) or visual question answering (VQA Zitnick and Parikh (2013)). All of these tasks involve indirectly some notion of intuitive physics. Our proposed test directly measures physical understanding in a task- and model-agnostic way.

events, even if these videos are entirely novel. However, the model should give large prediction errors when some unlikely or impossible event happens. In other words, impossible events have a zero probability in the real world, so a model trained only with possible events should be able to generalize to other possible events, while rejecting impossible ones.

This is directly inspired by the "violation of expectation" (VOE) paradigm in cognitive psychology, whereby infants or animals are presented with real or virtual animated 3D scenes which may contain a physical impossibility. The "surprise" reaction to the physical impossibility is measured through looking time or other physiological measures, and is taken to reflect a violation of it's internal predictions Baillargeon et al. (1985). Similarly, our evaluation requires systems to output a scalar variable upon the presentation of a video clip, which we will call a '*plausibility score*' (it could be a log probability, an inverse reconstruction error, etc). We expect the

plausibility score to be lower for clips containing the violation of a physical principle than for matched clips with no violation. By varying the nature of the physical violation, one can probe different types of physical laws: object permanence (objects don't pop in and out of existence), shape constancy (objects keep their shapes), spatio-temporal continuity (trajectories of objects are continuous) [1]. These three physical laws form the three blocks of IntPhys2019.

As in infant's experiments, our tests are constructed in well matched sets of clips, i.e., the possible versus impossible clips differ minimally, in order to minimize the possibility of dataset biases, but are quite varied, to maximize the difficulty of solving the test through simple heuristics. Three additional advantages of this method are that (1) they provide directly interpretable results (as opposed to a prediction error, or a composite score reflecting an entire pipeline), (2) they enable to probe generalization for difficult cases outside of the training distribution, which is useful for systems that are intended to work in the real world, and (3) they enable for rigorous human-machine comparison, which is important in order to quantify how far are artificial system in matching human intuitive physical understanding.

Our tests have also limits, which are the flip side of their advantage: They measure intuitive physics as looked through the prediction errors of a system, but do not measure how well a system might be able to use this kind of understanding. For instance, an end-to-end VQA system may have superb physical understanding (as measured by VOE) but fail miserably in connecting it with language. In this sense, VOE should be viewed as a diagnostic tool, a kind if *unit testing* for physics that needs to be combined with other measures to fully evaluate end-to-end systems. Similarly these tests do not exhaustively probe for all aspects of intuitive physics, but rather break it down into a small set of basic concepts tested one at a time. Here again, unit testing does not guarantee that an entire system will work correctly, but it helps to understand what happens when it does not.

[1] This has a direct parallel in 'black box' evaluation of language models in NLP. Language models are typically trained with a future prediction objective (predicting future characters or words conditioned on past ones). However, instead of evaluating theses models directly on the loss function or derivatives like perplexity, an emerging research direction is to the models on artificially constructed sentences that violate certain grammatical rules (like number agreement) measure the ability of the system to detect these violations Linzen et al. (2016)

This paper is structured as follows. In Section 3.2, we present the IntPhys Benchmark, which tests for 3 basic concepts of intuitive physics in a VOE paradigm. In Section 3.3, we describe two baseline systems which are trained with a self-supervised frame prediction objective on the training set, and in Section 3.4 we analyse their performance compared to that of human participants. In Section 3.5 we present related work and conclude in Section 3.6 by discussing the next steps in extending this approach to more intuitive physics concepts and how they could be augmented to incorporate testing of decision and planning.

3.2 Structure of the IntPhys benchmark

IntPhys is a benchmark designed to address the evaluation challenges for intuitive physics in vision systems. It can be run on any of machine vision system (captioning and VQA systems, systems performing 3D reconstruction, tracking, planning, etc), be they engineered by hand or trained using statistical learning, the only requirement being that the tested system should output a scalar for each test video clip reflecting the *plausibility* of the clip as a whole. Such a score can be derived from prediction errors, or posterior probabilities, depending on the system.

In this release we have implemented tests for three basic concepts of the physics of macroscopic solid objects: object permanence, shape constancy, spatio-temporal continuity. Each of these concepts are tested in a series of controlled possible and impossible clips, which are presented without labels, and for which models have to return a plausibility score. The evaluation is done upon submission of these scores in CodaLab, and the results are automatically presented in a leaderboard. This benchmark also contains a training set of videos with random object interactions, in a similar environment as for the test set. This can be used either to train predictive systems or to conduct domain adaptation for systems trained on other datasets (live videos, virtual environments, robots). Obviously, the training set only contains physically possible events.

This benchmark will be the first evaluation of the DARPA Machine Common

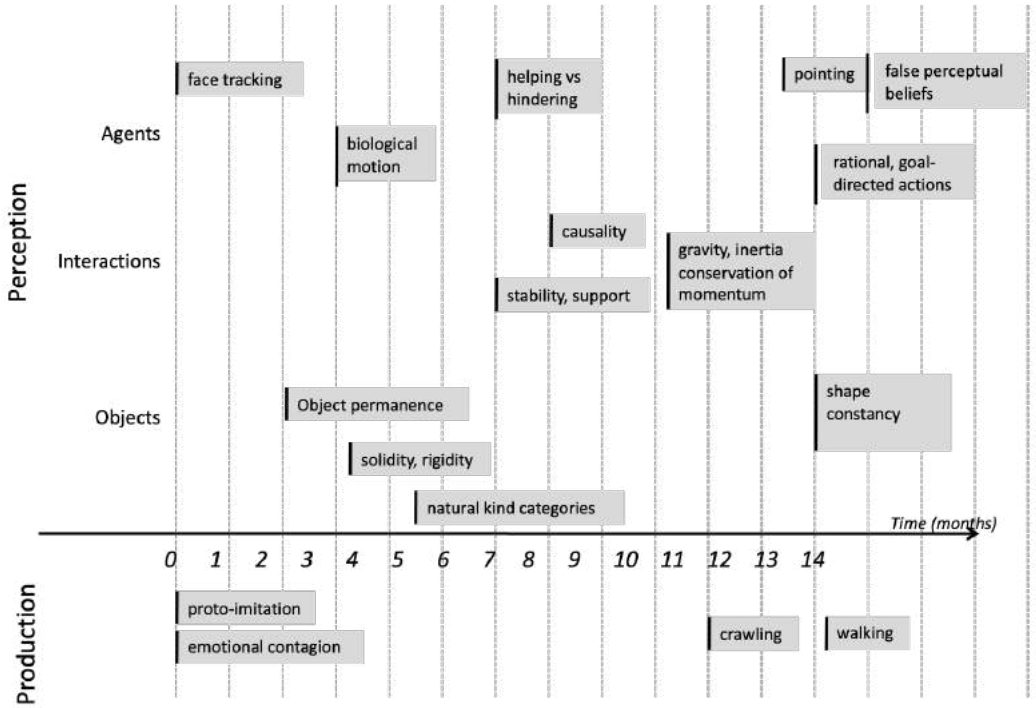


Figure 3.2: Landmark of intuitive physics acquisition in infants. Each box is an experiment showing a particular ability at a given age.

Sense project [2], a research program seeking to address the challenge of machine common sense to enable systems to understand new situations and monitor the reasonableness of their actions. This will allow all teams involved in the program to evaluate their systems on a common ground.

3.2.1 Three basic concepts of intuitive physics

Behavioral work on intuitive physics in infants and animal define a number of core conceptual components which can be tested experimentally using VOE Baillargeon and Carey (2012). Figure 3.2 shows a number of different landmarks in infants. Here, we have selected three of the most basic components and turned them into three test

[2] www.darpa.mil/program/machine-common-sense

Table 3.1: List of the conceptual blocks of the Intuitive Physics Framework.

Block Name	Physical principles	Computational challenge
O1. Object permanence	Objects don't pop in and out of existence	Occlusion-resistant object tracking
O2. Shape constancy	Objects keep their shapes	Appearance-robust object tracking
O3. Spatio-temporal continuity	Trajectories of objects are continuous	Tracking/predicting object trajectories

blocks (see 3.1), each one corresponding to a core principle of intuitive physics, and each raising its particular machine vision challenge. The first two blocks are related to the conservation through time of intrinsic properties of objects. Object permanence (O1), corresponds to the fact that objects continuously exist through time and do not pop in or out of existence. This turns into the computational challenge of tracking objects through occlusion. The second block, shape constancy (O2) describes the tendency of rigid objects to preserve their shape through time. This principle is more challenging than the preceding one, because even rigid objects undergo a change in appearance due to other factors (illumination, distance, viewpoint, partial occlusion, etc.). The final block (O3) relate to object's trajectories, and posit that each object's motion has to be continuous through space and time (an object cannot teleport from one place to another). This principle is distinct from object permanence and requires a to incorporate smoothness constraints on the tracking of objects (even if they are not visible). Future releases of the Benchmark will continue adding progressively more complex scenarios inspired by Figure 3.2, including object interactions and agent motion.

3.2.2 Pixels matched quadruplets

An important design principle of our evaluation framework relates to the organization of the possible and impossible movies in extremely well matched sets to minimize the existence of low level biases. This is illustrated in Figure 3.3 for object permanence. We constructed matched sets comprising four movies, which contain an initial scene at time t_1 (either one or two objects), and a final scene at time t_2 (either one or two objects), separated by a potential occlusion by a screen which is

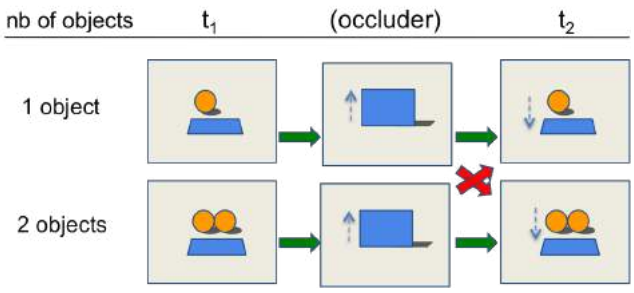


Figure 3.3: Illustration of the minimal sets design with object permanence. Schematic description of a static condition with one vs. two objects and one occluder. In the two possible movies (green arrows), the number of objects remains constant despite the occlusion. In the two impossible movies (red arrows), the number of objects changes (goes from 1 to 2 or from 2 to 1).

raised and then lowered for a variable amount of time. At its maximal height, the screen completely occludes the objects so that it is impossible to know, in this frame, how many objects are behind the occluder.

The four movies are constructed by combining the two possible beginnings with the two possible endings, giving rise to two possible ($1\rightarrow1$ and $2\rightarrow2$) and two impossible ($1\rightarrow2$ and $2\rightarrow1$) movies. Importantly, across these 4 movies, the possible and impossible ones are made of frames with the exact same pixels, the only factor distinguishing them being the temporal coherence of these frames. To verify this, we compute the `SHA256` hash of frames for both possible and impossible events, sort them in lexicographic order, and make sure the two lists match [3]. Such a design is intended to make it difficult for algorithms to use cheap tricks to distinguish possible from impossible movies by focusing on low level details, but rather requires models to focus on higher level temporal dependencies between frames.

[3] This experiment can but found here: `www.github.com/rronan/IntPhys-verify-quadruplets`

3.2.3 Parametric task complexity

Our second design principle is that in each block, we vary the stimulus complexity in a parametric fashion. In the case of the object permanence block, for instance, stimulus complexity can vary according to three dimensions. The first dimension is whether the change in number of objects occurs in plain view (*visible*) or hidden behind an occluder (*occluded*). A change in plain view is evidently easier to detect whereas a hidden change requires an element of short term memory in order to keep a trace of the object's through time. The second dimension is the complexity of the object's motion. Tracking an immobile object is easier than if the object has a complicated motion; we introduce three levels of motion complexity (static, dynamic 1, and dynamic 2). The third dimension is the number of objects involved in the scene. This tests for the attentional capacity of the system as defined by the number of objects it can track simultaneously. Manipulating stimulus complexity is important to establish the limit of what a vision system can do, and where it will fail. For instance, humans are well known to fail when the number of objects to track simultaneously is greater than four Pylyshyn and Storm (1988). In total, a given block contains 2 by 3 by 3, ie, 18 different scenarios varying in difficulty (see Tables 3.5).

3.2.4 Procedurally generated variability

Our final design principle is that each scenario within each block is procedurally generated in 200 examplars with random variations in objects shapes and textures, distances, trajectories, occluder motion and position of the camera. This is to minimize the possibility of focusing on only certain frames or parts of the screen to solve the task. Note that the dynamic 2 condition contains two violations instead of one. These violations are inverses of one another, such that the first and last segment of the impossible video clips are compatible with with the absence of any violation in the central part of the video (for instance, the initial and final number of objects is the same, but varies in the middle of the clip). This ensures that physical violations occur in unpredictable moments in a video clip.

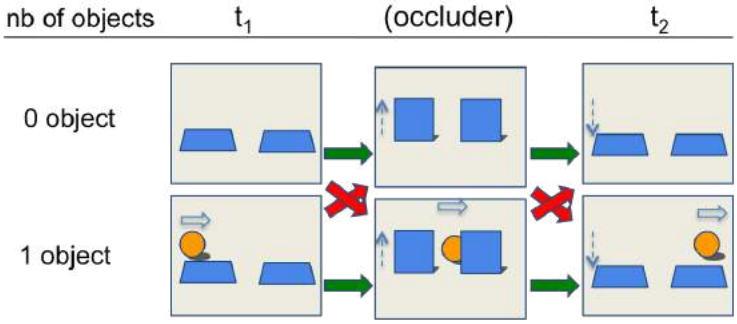


Figure 3.4: Illustration of the 'dynamic 2' condition. In the two possible movies (green arrows), the number of objects remains constant despite the occlusion. In the two impossible movies (red arrows), the number of objects changes temporarily (goes from 0 to 1 to 0 or from 1 to 0 to 1).

3.2.5 The possible versus impossible discrimination metric

Our evaluation metrics depend on the system's ability to compute a *plausibility score* $P(x)$ given a movie x. Because the test movies are structured in N matched k-uplets (in Figure 3.3, $k = 4$) of positive and negative movies $S_{i=1..N} = \{Pos_i^1..Pos_i^k, Imp_i^1..Imp_i^k\}$, we derive two different metrics. The *relative* error rate L_R computes a score within each set. It requires only that within a set, the positive movies are more plausible than the negative movies.

$$L_R = \frac{1}{N} \sum_i \mathbb{1}_{\sum_j P(Pos_i^j) < \sum_j P(Imp_i^j)} \tag{3.1}$$

The absolute error rate L_A requires that globally, the score of the positive movies is greater than the score of the negative movies. It is computed as:

$$L_A = 1 - AUC(\{i, j; P(Pos_i^j)\}, \{i, j; P(Imp_i^j)\}) \tag{3.2}$$

Where AUC is the Area Under the ROC Curve, which plots the true positive rate against the false positive rate at various threshold settings.

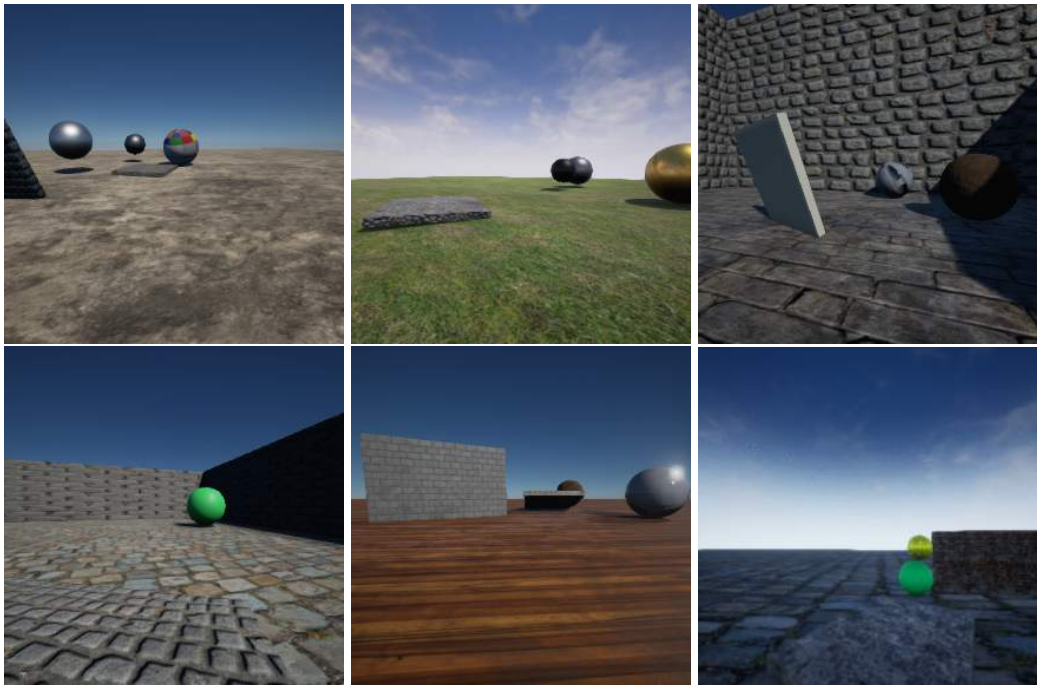


Figure 3.5: Examples of frames from the training set.

3.2.6 Implementation

The video clips in IntPhys are constructed with Unreal Engine 4.0 ((UnrealEnginePython 4.19; See Figure 3.5 for some examples). They are accessible in `www.intphys.com`.

The training set

The training set contains a large variety of objects interacting one with another, occluders, textures, etc. It is composed of 15K videos of possible events (around 7 seconds each at 15fps), totalling 21 hours of videos. There are no video of impossible events, but the training set contains the objects and occluders presented in the test set. Each video is delivered as stacks of raw image (288 x 288 pixels), totalling 157Gb of uncompressed data. We also release the source code for data generation, allowing users to generate a larger training set if desired.

The dev and test sets

As described above, each of the three blocks contain 18 different scenario. In the dev set, each scenario is instantiated by 20 different renderings resulting in 360 movies per block (30 min, 3.7Gb). In the test set, a scenario has 200 different renderings of these scenarios, resulting in a total of 3600 movies per block (5h,37Gb). All of the objects and textures of the dev and test sets are present in the training set.

The purpose of the dev set released in IntPhys V1.0 is to help in the selection of an appropriate plausibility score, and in the comparison of various architectures (hyper-parameters), but it should *not* serve to train the model's parameters (this should be done only with the training set). This is why the dev set is kept intentionally small. The test set has more statistical power and enables a fine grained evaluation of the results across the different movie subtypes. Video examples of each blocks are available on the project page `www.intphys.com`.

Metadata

Even though the spirit of IntPhys is the unsupervised learning of intuitive physics, we do provide in the test set additional information which may help the learner. The first one is the depth field for each image. This is not unreasonable, given that in infants, stereo vision and motion cues could provide an approximation of this information Fox et al. (1980). The second one is object instance segmentation masks, which are helpful to recover abstract object positions but only provide local low-level information. Importantly, these masks are not linked to a specific object ID, and are randomly shuffled at each time frame. Linking instance segmentation masks to unique object IDs through time is indeed part of the object permanence problem that systems are supposed to solve. Similarly, if an object is partly occluded and appears as two pieces of object the two pieces will receive a different instance mask.

In the train set, we do provide additional metadata about the ground truth 3D position of each object, the position of the camera, and the link between object IDs and instance masks. These metadata are not present in the dev or test sets.

Submission procedure

For each movie in the dev or test set, the model should issue a scalar plausibility score. This number together with the movie ID is then fed to the evaluation software which outputs two tables of results, one for the absolute score and the other for the relative score.

The evaluation software is provided for the dev set, but not the test set. For evaluating on the test set, participants are invited to submit their system and results on CodaLab (see `www.intphys.com`) and their results will be registered and time-stamped on the website leaderboard.

3.3 Two baseline learning models

In this section, we present two learning systems which attempt to learn intuitive physics in an unsupervised/self-supervised observational setting. One can imagine an agent who only sees physical interactions between objects seen from a first-person perspective, but cannot move nor interact with them. Arguably, this is a much more impoverished learning situation than that faced by infants, who can explore and interact with their environment, even with the limited motor abilities of their first year of life. It is however interesting to establish how far one can get with such simplified inputs, which are easy to gather in abundant amounts in the real world with video cameras. In addition, this enables an easier comparison between models, because they all get the same training data.

In a setup like this, a rich source of learning information resides in the temporal dependencies between successive frames. Based on the literature on next frame prediction, we propose two neural network models, trained on a future frame objective. Our first model has a CNN encoder-decoder structure and the second is a conditional Generative Adversarial Network (GAN, Goodfellow et al. (2014)), with a similar structure as DCGAN Radford et al. (2015). For both model architectures, we investigate two different training procedures: in the first, we train models to predict short-future images with a prediction span of 5 frames; in the second, we predict long-future images with a prediction span of 35 frames.

Preliminary work with predictions at the pixel level revealed that our models

failed at predicting convincing object motions, especially for small objects on a rich background. For this reason, we switched to computing predictions at a higher level, using object masks. We use the metadata provided in the benchmark training (see section 4.6.2) set to train a semantic mask Deep Neural Network (DNN). This DNN uses a resnet-18 pretrained on Imagenet to extract features from the image, from which a deconvolution network is trained to predict the semantic mask (distinguishing three types of entities: background, occluders and objects). We then use this mask as input to a prediction component which predicts future masks based on past ones.

To evaluate these models on our benchmark, our system needs to output a plausibility score for each movie. For this, we compute the prediction loss along the movie. Given past frames, a plausibility score for the frame f_t can be derived by comparing f_t with the prediction $\hat{f}_t$. Like in Fragkiadaki et al. (2015a), we use the analogy with an agent running an internal simulation ("visual imagination"); here we assimilate a greater distance between prediction and observation with a lower plausibility. In subsection 3.3.2 we detail how we aggregate the scores of all frames into a plausibility score for the whole video.

3.3.1 Models

Through out the movie, our models take as input two frames (f_{i_1}, f_{i_2}) and predict a future frame f_{target}. The prediction span is independent from the model's architecture and depends only on the triplets $(f_{i_1}, f_{i_2}, f_{target})$ provided during the training phase. Our two architectures are trained either on a short term prediction task (5 frames in the future), or a long term prediction task (35 frames). Intuitively, short-term prediction will be more robust, but long-term prediction will allow the model to grasp long-term dependencies and deal with long occlusions.

CNN encoder-decoder

We use a resnet-18 He et al. (2016) pretrained on Imagenet Russakovsky et al. (2015) to extract features from input frames (f_{i_1}, f_{i_2}). A deconvolution network is trained to predict the semantic mask of future frame f_{target} conditioned to these

features, using a L2 loss. See details in Table 3.2

Table 3.2: CNN for forward prediction (13941315 parameters). BN stands for batch-normalization.

Input frames 2 x 3 x 64 x 64
7 first layers of resnet-18 (pretrained, frozen weights) applied to each frame
Reshape 1 x 16384
FC 16384 $\rightarrow$ 512
FC 512 $\rightarrow$ 8192
Reshape 128 x 8 x 8
UpSamplingNearest(2), 3 x 3 Conv. 128 - 1 str., BN, ReLU
UpSamplingNearest(2), 3 x 3 Conv. 64 - 1 str., BN, ReLU
UpSamplingNearest(2), 3 x 3 Conv. 3 - 1 str., BN, ReLU
3 sigmoid Target mask

Generative Adversarial Network

As a second model, we propose a conditional generative adversarial network (GAN, Mirza and Osindero (2014)) that takes as input predicted semantic masks from frames (f_{i_1}, f_{i_2}), and predicts the semantic mask of future frame f_{target}. In this setup, the discriminator has to distinguish between a mask predicted from f_{target} directly (*real*), and a mask predicted from past frames (f_{i_1}, f_{i_2}). Like in Denton et al. (2016), our model combines a conditional approach with a similar structure as of DCGAN Radford et al. (2015). At test time, we derive a plausibility score by computing the conditioned discriminator's score for every conditioned frame. This is a novel approach based on the observation that the optimal discriminator D computes a score for x of

$$D(x) = \frac{P_{data}(x)}{P_G(x) + P_{data}(x)} \tag{3.3}$$

For non-physical events $\hat{x}$, $P_{data}(\hat{x}) = 0$; therefore, as long as $P_G(\hat{x}) > 0$, $D(\hat{x})$ should be 0 for non-physical events, and $D(x) > 0$ for physical events x. Note that this is a strong assumption, as there is no guarantee that the generator will ever have support at the part of the distribution corresponding to impossible videos. The generator and discriminator are detailed in Table 3.3 and 3.4, respectively.

Table 3.3: Generator G (14729347 parameters). SFConv stands for spatial full convolution and BN stands for batch-normalization.

Input masks 2 x 3 x 64 x 64	
4 x 4 conv 64 - 2 str., BN, ReLU	
4 x 4 conv 128 - 2 str., BN, ReLU	
4 x 4 conv 256 - 2 str., BN, ReLU	Noise $\in \mathbf{R}^{100}$ $\sim$ Unif$(-1, 1)$
4 x 4 conv 512 - 2 str., BN, ReLU	
4 x 4 conv 512, BN, ReLU	
stack input and noise	
4 x 4 SFConv. 512 - 2 str., BN, ReLU	
4 x 4 SFConv. 256 - 2 str., BN, ReLU	
4 x 4 SFConv. 128 - 2 str., BN, ReLU	
4 x 4 SFConv. 64 - 2 str., BN, ReLU	
4 x 4 SFConv. 3 - 2 str., BN, ReLU	
3 sigmoid Target mask	

Training Procedure

We separate 10% of the training dataset to control the overfitting of our forward predictions. All our models are trained using `Adam` (Kingma and Ba (2014)). For the CNN encoder-decoder we use Adam's default parameters and stop the training after one epoch. For the GAN, we use the same hyper-parameters as in Radford et al. (2015): we set the generator's learning rate to $8e-4$ and discriminator's learning rate to $2e-4$. On the short-term prediction task, we train the GAN for 1 epoch; on the long-term prediction task we train it for 5 epochs. Learning rate decays are set to 0 and $beta1$ is set to 0.5 for both generator and discriminator.

Table 3.4: Discriminator D (7629698 parameters). BN stands for batch-normalization.

history 2 x 3 x 64 x 64	input 3 x 64 x 64
Reshape 3 x 3 x 64 x 64	
4 x 4 convolution 512 - 2 strides, BN, LeakyReLU	
4 x 4 convolution 254 - 2 strides, BN, LeakyReLU	
4 x 4 convolution 128 - 2 strides, BN, LeakyReLU	
4 x 4 convolution 64 - 2 strides, BN, LeakyReLU	
4 x 4 convolution 5 - 2 strides, BN, LeakyReLU	
fully-connected layer	
1 sigmoid	

The code for all our experiments is available on `www.github.com/rronan/IntPhys-Baselines`.

3.3.2 Video Plausibility Score

From forward models presented above, we can compute a plausibility score for every frame f_{target}, conditioned on previous frames (f_{i_1}, f_{i_2}). However, because the temporal positions of impossible events are not given, we must decide of a score for a video, given the scores of all its conditioned frames. An impossible event can be characterized by the presence of one or more impossible frame(s), conditioned to previous frames. Hence, a natural approach to compute a video plausibility score is to take the minimum of all conditioned frames' scores:

$$\text{Plaus}(v) = \min_{(f_{i_1}, f_{i_2}, f_{target}) \in v} \text{Plaus}(f_{target} | f_{i_1}, f_{i_2}) \tag{3.4}$$

where v is the video, and $(f_{i_1}, f_{i_2}, f_{target})$ are all the frame triplets in v, as given in the training phase.

3.3.3 Results

Block O1

Short-term prediction The first training procedure is a short-term prediction task; it takes as input frames f_{t-2}, f_t and predicts f_{t+5}, which we note $(f_{t-2}, f_t) \rightarrow f_{t+5}$ in the following. We train the two architectures presented above on short-term prediction task and evaluate them on the test set. For the relative classification task, CNN encoder-decoder has an error rate of 0.09 when impossible events are visible and 0.49 when they are occluded. The GAN has an error rate of 0.15 when visible and 0.48 when occluded. For the absolute classification task, CNN encoder-decoder has a L_A (see eq. 3.2) of 0.33 when impossible events are visible and 0.50 when they are occluded. The GAN has a L_A of 0.38 when visible and 0.50 when occluded. Results are detailed in Supplementary Materials (Tables 1, 2, 3, 4).
We observe that our short-term prediction models show good performances when the impossible events are visible, especially on the relative classifications task. However they perform poorly when the impossible events are occluded. This is easily explained by the fact that they have a prediction span of 5 frames, which is usually lower than the occlusion time. Hence, these models don't have enough "memory" to catch occluded impossible events.

Long-term prediction The second training procedure consists in a long-term prediction task: $(f_{t-5}, f_t) \rightarrow f_{t+35}$. For the relative classification task, CNN encoder-decoder has an error rate of 0.07 when impossible events are visible and 0.52 when they are occluded. The GAN has an error rate of 0.17 when visible and 0.48 when occluded. For the absolute classification task, CNN encoder-decoder has a L_A of 0.37 when impossible events are visible and 0.50 when they are occluded. The GAN has a L_A of 0.40 when visible and 0.50 when occluded. Results are detailed in Supplementary Materials (Tables 5, 6, 7, 8). As expected, long-term models perform better than short-term models on occluded impossible events. Moreover, results on absolute classification task confirm that it is way more challenging than the relative classification task. Because some movies are more complex than others, the average score of each quadruplet of movies may vary a lot. It results in cases where one

model returns a higher plausibility score to an impossible movie $M_{\{\text{imp, easy}\}}$ from an easy quadruplet than to a possible movie $M_{\{\text{pos, complex}\}}$ from a complex quadruplet.

Aggregated model On the relative classification task, the aggregated CNN encoder-decoder has an error rate of 0.07 when impossible events are visible and 0.52 when they are occluded. For the absolute classification task, CNN encoder-decoder has a L_A of 0.37 when impossible events are visible and 0.50 when they are occluded. Results are detailed in Figures 3.6 and Supplementary Materials (Tables 9, 10).

Block O2

Short-term prediction For the first training procedure $(f_{t-2}, f_t) \rightarrow f_{t+5}$: CNN encoder-decoder has an relactive classification error rate of 0.16 when impossible events are visible and 0.49 when they are occluded. The GAN has an error rate of 0.30 when visible and 0.52 when occluded. For the absolute classification task, CNN encoder-decoder has a L_A of 0.40 when impossible events are visible and 0.50 when they are occluded. The GAN has a L_A of 0.43 when visible and 0.50 when occluded. Results are detailed in Supplementary Materials (Tables 11, 12, 13, 14).

Long-term prediction For the second training procedure $(f_{t-5}, f_t) \rightarrow f_{t+35}$: the CNN encoder-decoder has an error rate of 0.11 when impossible events are visible and 0.52 when they are occluded. The GAN has an error rate of 0.31 when visible and 0.50 when occluded. For the absolute classification task, CNN encoder-decoder has a L_A of 0.43 when impossible events are visible and 0.50 when they are occluded. The GAN has a L_A of 0.33 when visible and 0.50 when occluded. Results are detailed in Supplementary Materials (Tables 15, 16, 17, 18).

Aggregated model On the relative classification task, the aggregated CNN encoder-decoder has an error rate of 0.11 when impossible events are visible and

0.52 when they are occluded. For the absolute classification task, CNN encoder-decoder has a L_A of 0.43 when impossible events are visible and 0.50 when they are occluded. Results are detailed in Figures 3.6 and Supplementary Materials (Tables 19, 20).

Block O3

Short-term prediction For the first training procedure $(f_{t-2}, f_t) \rightarrow f_{t+5}$: CNN encoder-decoder has an relactive classification error rate of 0.28 when impossible events are visible and 0.49 when they are occluded. The GAN has an error rate of 0.26 when visible and 0.48 when occluded. For the absolute classification task, CNN encoder-decoder has a L_A (see eq. 3.2) of 0.40 when impossible events are visible and 0.50 when they are occluded. The GAN has a L_A of 0.42 when visible and 0.50 when occluded. Results are detailed in Supplementary Materials (Tables 21, 22, 23, 24).

Long-term prediction For the second training procedure $(f_{t-5}, f_t) \rightarrow f_{t+35}$: the CNN encoder-decoder has an error rate of 0.32 when impossible events are visible and 0.51 when they are occluded. The GAN has an error rate of 0.34 when visible and 0.52 when occluded. For the absolute classification task, CNN encoder-decoder has a L_A of 0.46 when impossible events are visible and 0.50 when they are occluded. The GAN has a L_A of 0.44 when visible and 0.50 when occluded. Results are detailed in Supplementary Materials (Tables 25, 26, 27, 28).

Aggregated model On the relative classification task, the aggregated CNN encoder-decoder has an error rate of 0.32 when impossible events are visible and 0.51 when they are occluded. For the absolute classification task, CNN encoder-decoder has a L_A of 0.46 when impossible events are visible and 0.50 when they are occluded. Results are detailed in Figures 3.6 and Supplementary Materials (Tables 29, 30).

As expected, we observe that models' performance decrease when impossible

events are occluded. This enlightens the difficulty to perform long-term predictions in videos. We also observe that their performances vary with the types of impossible events tested. Results are the highest when testing presence / absence of object, and the lowest when testing the temporal continuity of trajectories.

3.3.4 Results from other works.

Other works have reported results on IntPhys. Among those works, Smith et al. (2019) and Riochet et al. (2020b) both integrate a visual and a physics module. Their visual modules allows to parse the scene and in an object representation, while the physics modules use this representation to infer physical properties and predict trajectories of objects. While Smith et al. (2019) use an hand-crafted stochastic physics engine, Riochet et al. (2020b) train a graph neural network on observations from our training set. Smith et al. (2019) evaluate on block $O1$ only, with a reported average relative score of 0.27. Riochet et al. (2020b) report relative scores of 0.12, 0.21, and 0.37 on blocks $O1$, $O2$, and $O3$ respectively. The performances of those works, compared to our pixel-based models, tend to show the benefits of hybrid architectures combining visual modules and object-based physics models.

3.4 Human Judgements Experiment

To give a second reference to evaluate physical understanding in models, and provide a good description of human performance on this benchmark, we presented the 3600 videos from each block to human participants using Amazon Mechanical Turk. Participants were first presented 8 examples of possible scenes from the training set, some simple, some more complex. They were told that some of the test movies were incorrect or corrupted, in that they showed events that could not possibly take place in the real world (without specifying how). Participants were each presented with 40 randomly selected videos, and were asked to score them from 1 (most implausible) to 6 (most plausible). They completed the task in about 7 minutes, and were paid $1. A total of 540 persons participated, such that every video tested was seen by 2 different participants. A mock sample of the AMT test is available on `http://129.199.81.135/naive_physics_experiment`.

Table 3.5: Average error rate on plausibility judgments collected in humans using MTurk for IntPhys test set. * *EDIT July 2021: these experiments present a flaw and the results do not accurately reflect human judgement on block O3/Occluded/Dynamic. For more information, please contact ronan.riochet@inria.fr.*

	Visible				Occluded			
Type of scene	1 obj.	2 obj.	3 obj.	Total	1 obj.	2 obj.	3 obj.	Total
	Block O1							
Static	0.13	0.14	0.09	0.12 ($\pm 0,018$)	0.32	0.34	0.28	0.31 (± 0.026)
Dynamic (1 violation)	0.15	0.29	0.27	0.24 ($\pm 0,024$)	0.24	0.30	0.33	0.29 (± 0.026)
Dynamic (2 violations)	0.14	0.20	0.23	0.19 (± 0.022)	0.28	0.26	0.36	0.30 (± 0.026)
Total	0.14	0.21	0.20	0.18 (± 0.013)	0.28	0.30	0.32	0.30 (± 0.015)
	Block O2							
Static	0.13	0.18	0.15	0.16 (± 0.021)	0.22	0.33	0.28	0.28 (± 0.025)
Dynamic (1 violation)	0.29	0.24	0.27	0.27 (± 0.025)	0.29	0.35	0.29	0.31 (± 0.026)
Dynamic (2 violations)	0.21	0.27	0.26	0.24 (± 0.024)	0.32	0.32	0.29	0.31 (± 0.026)
Total	0.21	0.23	0.23	0.22 (± 0.014)	0.28	0.33	0.29	0.30 (± 0.015)
	Block O3							
Static	0.29	0.32	0.27	0.29 (± 0.026)	0.36	0.36	0.45	0.39 (± 0.028)
Dynamic (1 violation)	0.28	0.33	0.30	0.30 (± 0.026)	0.49 *	0.55*	0.49*	0.51* (± 0.028)
Dynamic (2 violations)	0.23	0.23	0.26	0.24 (± 0.024)	0.47*	0.53*	0.55*	0.52* (± 0.028)
Total	0.27	0.29	0.28	0.28 (± 0.015)	0.44*	0.48*	0.50*	0.47* (± 0.016)

The average error rates were computed across condition, number of objects and visibility and are shown in Tables 3.5. In general, observers missed violations more often when the scene was occluded; we observe error rates going from 18% (visible) to 30% (occluded) for block $O1$, from 22% (visible) to 30% (occluded) for block $O2$, from 28% (visible) to 47% (occluded) for block $O3$. An interesting result is that the score of humans on block $O3$ is close to chance when objects are occluded. This shows that humans have trouble to detect changes in velocity of objects, when these changes occur when the object is occluded. We also observe an increase in error going from static to dynamic 1 (one occlusion) and from dynamic 1 to dynamic 2 (two occlusions), but this pattern was only consistently observed in the occluded condition. For visible scenario, the dynamic 1 appeared more difficult than the dynamic 2. This was probably due to the fact that when objects are visible, the dynamic 2 impossible scenarios contain two local discontinuities and are therefore easier to spot than when one discontinuity only is present. When the discontinuities occurred behind the occluder, the pattern of difficulties was reversed, presumably because participants started using heuristics, such as checking that the number of objects at the beginning

is the same as at the end, and therefore missed the intermediate disappearance of an object.

These results suggest that human participants are not responding according to the gold standard laws of physics due to limitations in attentional capacity - and this, even though the number of objects to track is below the theoretical limit of 4 objects. The performance of human observers can thus serve as a reference besides ground truth, especially for systems intended to model human perception.

Interestingly, we observe similar patterns of performance between models and humans (see Figures 3.6), with increasing error rates from blocks $O1$ to $O3$. As expected, both humans and models show higher error rates when the considered impossible event is occluded.

3.5 Related work

The modeling of intuitive physics has been addressed mostly through systems trained with some form of future prediction as a training objective. Some studies have investigated models for predicting the stability and forward modeling the dynamics of towers of blocks (Battaglia et al. (2013); Lerer et al. (2016); Zhang et al. (2016); Li et al. (2016); Mirza et al. (2017); Li et al. (2017)). Battaglia et al. (2013) proposes a model based on an intuitive physics engine, Lerer et al. (2016) and Li et al. (2016) follow a supervised approach using Convolutional Neural Networks (CNNs), Zhang et al. (2016) makes a comparison between simulation-based models and CNN-based models, Mirza et al. (2017) improves the predictions of a CNN model by providing it with a prediction of a generative model. In Wu et al. (2016), authors propose a dataset and model to estimate object properties from visual inputs. In Mathieu et al. (2015), authors propose different feature learning strategies (multi-scale architecture, adversarial training method, image gradient difference loss function) to predict future frames in raw videos.

Other models use more structured representation of objects to derive longer-term predictions. In Battaglia et al. (2016) and Chang et al. (2016), authors learn objects dynamics by modelling their pairwise interactions and predicting the resulting objects states representation (e.g. position / velocity / object intrinsic properties) . In Watters

et al. (2017), Fraccaro et al. (2017) and Ehrhardt et al. (2017a) authors combine factored latent object representations, object centric dynamic models and visual encoders. Each frame is parsed into a set of object state representations, which are used as input of a dynamic model. In Fraccaro et al. (2017) and Ehrhardt et al. (2017a), authors use a visual decoder to reconstruct the future frames, allowing the model to learn from raw (though synthetic) videos.

Regarding evaluation and benchmarks, apart from frame prediction datasets, which are not strictly speaking about intuitive physics, one can distinguish the Visual Newtonian Dynamics (VIND) dataset which includes more than 6000 videos with bounding boxes on key objects across frames, and annotated with a 3D plane which would most closely fit the object trajectory Mottaghi et al. (2016). Bakhtin et al. (2019) and Allen et al. (2020) propose two benchmarks for physical reasoning involving action-reward setups in a 2D environments. There are also two recent datasets proposed by a DeepMind team Piloto et al. (2018), and a MIT team Smith et al. (2019). These last datasets seem very similar to ours, they are inspired by the developmental literature and based on the violation of expectation principles and are structured around similar intuitive physics blocks. Piloto et al. (2018) have 3 blocks similar to ours (object permanence, shape constancy, continuity) and two other ones on solidity and containment. Differently to our work, they have one training set per block with *consistent examples*: explicitly designed to be similar as possible videos in the test set (with higher variability), and *controls*: designed to mitigate biases for these block. Like our work, Smith et al. (2019) design one single training set, where object motion are not specifically constrained. In our work, two differences emerge from Piloto et al. (2018); Smith et al. (2019): our dataset is better matched in terms of quadruplets of clips controlled at the level of the pixels, and our dataset has a factorial manipulation of scene and movement complexity.It would be interesting to explore the possibility to merge these datasets, as well as add more blocks in order to increase the diversity and coverage of the physical phenomena.

3.6 Discussion

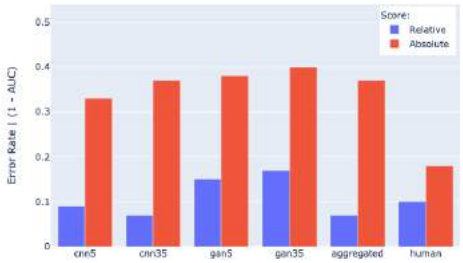
We presented IntPhys, a benchmark for measuring intuitive physics in artificial vision systems inspired by research on conceptual development in infants. To pass the benchmark, a system is asked to return a plausibility score for each video clip. The system's performance is assessed by measuring its ability to discriminate possible from impossible videos illustrating several types of physical principles. Naive humans were tested on the same dataset, to give an idea of what performance could be expected by a good model. These results show error rates increasing with the presence of occlusion, but not with number of objects. This is congruent with data showing that humans can track up to three objects simultaneously. We presented two unsupervised learning models based on semantic masks, which learn from a training set only composed of physically plausible clips, and are tested on the same block as the humans.

The computational system generally performed poorly compared to humans but obtained above chance performance in the visible cases using a mask prediction task. The relative success of the semantic mask prediction system compared to what we originally found with pixel-based systems indicates that operating at a more abstract level is a worthwhile pursuing strategy when it comes to modeling intuitive physics.

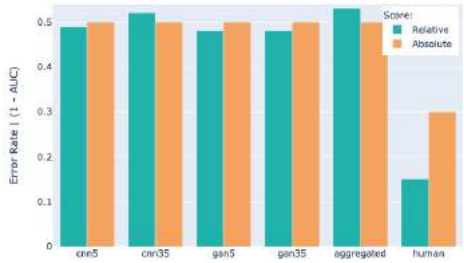
We report other works constructing this abstract representation in particular instance masks and object detection bounding boxes, showing better performances, especially in the presence of occlusions. In addition, enriching the training through embedding the learner in an interactive version of the environment could add more information for the learning of the physics of macroscopic objects.

In brief, the systematic way of constructing the IntPhys Benchmark shows that it is possible to adapt developmental paradigm in a machine learning setting, and that the resulting benchmark is a relatively challenging one. The three blocks that we present here could be extended to cover more aspects of object perception, including more difficult ones like interactions between objects, or prediction of trajectories of animated agents. As we discussed in the introduction, this benchmark only provides unit tests regarding the computation of prediction probabilities of object positions based on past frames. Further work will be needed to construct benchmarks testing how theses probabilities can be used by a system to make decision or plan

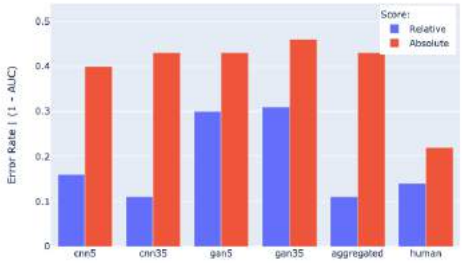
trajectories.



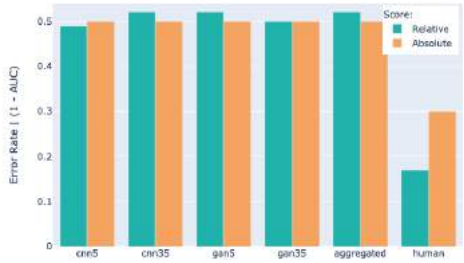
O1 visible



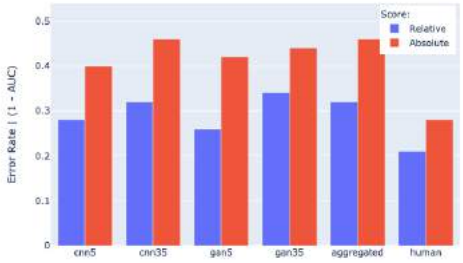
O1 occluded



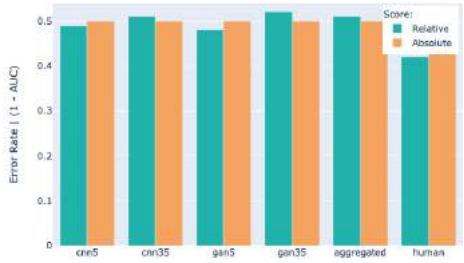
O2 visible



O2 occluded



O3 visible



O3 occluded

Figure 3.6: Results of our baselines on blocks $O1$, $O2$, $O3$, in cases where the impossible event occurs in the open (*visible*) or behind an occluder (*occluded*). Y-axis represents the losses L_R (see Equation 1) for the relative performance and L_A (see Equation 2) for the absolute performance.

3.7 Appendix

3.7.1 Model results (detailed)

Table 3.6: Block O1 | Model: CNN (short-term prediction task)

	Visible				Occluded			
Type of scene	1 obj.	2 obj.	3 obj.	Total	1 obj.	2 obj.	3 obj.	Total
	Relative classification (L_R)							
Static	0.00	0.00	0.00	0.00	0.49	0.52	0.41	0.47
Dynamic (1 violation)	0.00	0.22	0.27	0.17	0.51	0.47	0.49	0.49
Dynamic (2 violations)	0.00	0.13	0.20	0.11	0.50	0.50	0.49	0.50
Total	0.00	0.12	0.16	0.09	0.50	0.50	0.46	0.49
	Absolute classification (L_A)							
Static	0.15	0.17	0.19	0.17	0.50	0.50	0.49	0.50
Dynamic (1 violation)	0.32	0.44	0.47	0.41	0.50	0.50	0.50	0.50
Dynamic (2 violations)	0.33	0.43	0.47	0.41	0.50	0.50	0.50	0.50
Total	0.26	0.35	0.38	0.33	0.50	0.50	0.50	0.50

Table 3.7: Block O1 | Model: GAN (short-term prediction task)

	Visible				Occluded			
Type of scene	1 obj.	2 obj.	3 obj.	Total	1 obj.	2 obj.	3 obj.	Total
	Relative classification (L_R)							
Static	0.00	0.00	0.00	0.00	0.44	0.45	0.53	0.48
Dynamic (1 violation)	0.00	0.35	0.39	0.25	0.44	0.50	0.47	0.47
Dynamic (2 violations)	0.00	0.21	0.39	0.20	0.51	0.50	0.49	0.50
Total	0.00	0.18	0.26	0.15	0.46	0.48	0.50	0.48
	Absolute classification (L_A)							
Static	0.23	0.31	0.32	0.28	0.50	0.50	0.50	0.50
Dynamic (1 violation)	0.33	0.47	0.50	0.43	0.49	0.49	0.50	0.49
Dynamic (2 violations)	0.34	0.44	0.46	0.41	0.50	0.50	0.50	0.50
Total	0.30	0.41	0.43	0.38	0.49	0.50	0.50	0.50

Table 3.8: Block O1 | Model: CNN (long-term prediction task)

	Visible				Occluded			
Type of scene	1 obj.	2 obj.	3 obj.	Total	1 obj.	2 obj.	3 obj.	Total
	Relative classification (L_R)							
Static	0.00	0.00	0.00	0.00	0.52	0.55	0.51	0.53
Dynamic (1 violation)	0.00	0.13	0.22	0.12	0.49	0.53	0.48	0.50
Dynamic (2 violations)	0.00	0.06	0.20	0.09	0.53	0.48	0.60	0.54
	Absolute classification (L_A)							
Static	0.30	0.34	0.36	0.33	0.50	0.50	0.50	0.50
Dynamic (1 violation)	0.30	0.43	0.44	0.39	0.50	0.50	0.50	0.50
Dynamic (2 violations)	0.32	0.40	0.43	0.39	0.50	0.50	0.50	0.50
Total	0.31	0.39	0.41	0.37	0.50	0.50	0.50	0.50

Table 3.9: Block O1 | Model: GAN (long-term prediction task)

	Visible				Occluded			
Type of scene	1 obj.	2 obj.	3 obj.	Total	1 obj.	2 obj.	3 obj.	Total
	Relative classification (L_R)							
Static	0.00	0.01	0.00	0.00	0.41	0.58	0.57	0.52
Dynamic (1 violation)	0.00	0.28	0.45	0.24	0.39	0.56	0.54	0.50
Dynamic (2 violations)	0.01	0.29	0.46	0.25	0.43	0.46	0.40	0.43
Total	0.00	0.19	0.30	0.17	0.41	0.54	0.50	0.48
	Absolute classification (L_A)							
Static	0.26	0.33	0.37	0.32	0.50	0.50	0.50	0.50
Dynamic (1 violation)	0.36	0.46	0.49	0.44	0.49	0.50	0.50	0.50
Dynamic (2 violations)	0.35	0.47	0.48	0.43	0.50	0.50	0.50	0.50
Total	0.32	0.42	0.45	0.40	0.50	0.50	0.50	0.50

Table 3.10: Block O1 | Model: CNN aggregated

	Visible				Occluded			
Type of scene	1 obj.	2 obj.	3 obj.	Total	1 obj.	2 obj.	3 obj.	Total
	Relative classification (L_R)							
Static	0.00	0.00	0.00	0.00	0.52	0.55	0.51	0.53
Dynamic (1 violation)	0.00	0.13	0.22	0.12	0.49	0.53	0.48	0.50
Dynamic (2 violations)	0.00	0.06	0.20	0.09	0.53	0.48	0.60	0.54
Total	0.00	0.06	0.14	0.07	0.51	0.52	0.53	0.52
	Absolute classification (L_A)							
Static	0.30	0.34	0.36	0.33	0.50	0.50	0.50	0.50
Dynamic (1 violation)	0.30	0.43	0.44	0.39	0.50	0.50	0.50	0.50
Dynamic (2 violations)	0.32	0.40	0.43	0.39	0.50	0.50	0.50	0.50
Total	0.31	0.39	0.41	0.37	0.50	0.50	0.50	0.50

Table 3.11: Block O2 | Model: CNN (short-term prediction task)

	Visible				Occluded			
Type of scene	1 obj.	2 obj.	3 obj.	Total	1 obj.	2 obj.	3 obj.	Total
	Relative classification (L_R)							
Static	0.00	0.00	0.00	0.00	0.48	0.49	0.47	0.48
Dynamic (1 violation)	0.18	0.26	0.40	0.28	0.50	0.49	0.50	0.50
Dynamic (2 violations)	0.12	0.16	0.32	0.20	0.50	0.50	0.50	0.50
Total	0.10	0.14	0.24	0.16	0.49	0.49	0.49	0.49
	Absolute classification (L_A)							
Static	0.22	0.29	0.28	0.26	0.50	0.50	0.50	0.50
Dynamic (1 violation)	0.46	0.48	0.48	0.48	0.50	0.50	0.50	0.50
Dynamic (2 violations)	0.46	0.47	0.48	0.47	0.50	0.50	0.50	0.50
Total	0.38	0.42	0.42	0.40	0.50	0.50	0.50	0.50

Table 3.12: Block O2 | Model: GAN (short-term prediction task)

	Visible				Occluded			
Type of scene	1 obj.	2 obj.	3 obj.	Total	1 obj.	2 obj.	3 obj.	Total
	Relative classification (L_R)							
Static	0.00	0.00	0.00	0.00	0.54	0.55	0.47	0.52
Dynamic (1 violation)	0.44	0.38	0.47	0.43	0.56	0.52	0.54	0.54
Dynamic (2 violations)	0.38	0.52	0.51	0.47	0.50	0.50	0.48	0.49
Total	0.27	0.30	0.33	0.30	0.53	0.52	0.50	0.52
	Absolute classification (L_A)							
Static	0.29	0.30	0.32	0.30	0.50	0.50	0.50	0.50
Dynamic (1 violation)	0.49	0.49	0.49	0.49	0.50	0.50	0.50	0.50
Dynamic (2 violations)	0.48	0.49	0.50	0.49	0.50	0.50	0.50	0.50
Total	0.42	0.43	0.44	0.43	0.50	0.50	0.50	0.50

Table 3.13: Block O2 | Model: CNN (long-term prediction task)

	Visible				Occluded			
Type of scene	1 obj.	2 obj.	3 obj.	Total	1 obj.	2 obj.	3 obj.	Total
	Relative classification (L_R)							
Static	0.00	0.00	0.01	0.00	0.50	0.47	0.52	0.50
Dynamic (1 violation)	0.13	0.22	0.25	0.20	0.51	0.50	0.55	0.52
Dynamic (2 violations)	0.11	0.10	0.17	0.13	0.56	0.49	0.53	0.53
Total	0.08	0.11	0.14	0.11	0.52	0.49	0.54	0.52
	Absolute classification (L_A)							
Static	0.34	0.41	0.40	0.38	0.50	0.50	0.50	0.50
Dynamic (1 violation)	0.43	0.45	0.46	0.45	0.50	0.50	0.50	0.50
Dynamic (2 violations)	0.43	0.44	0.46	0.45	0.50	0.50	0.50	0.50
Total	0.40	0.43	0.44	0.43	0.50	0.50	0.50	0.50

Table 3.14: Block O2 | Model: GAN (long-term prediction task)

	Visible				Occluded			
Type of scene	1 obj.	2 obj.	3 obj.	Total	1 obj.	2 obj.	3 obj.	Total
	Relative classification (L_R)							
Static	0.02	0.02	0.00	0.01	0.49	0.55	0.53	0.52
Dynamic (1 violation)	0.35	0.42	0.54	0.44	0.50	0.40	0.45	0.45
Dynamic (2 violations)	0.44	0.51	0.53	0.50	0.56	0.44	0.53	0.51
Total	0.27	0.32	0.36	0.31	0.52	0.46	0.51	0.50
	Absolute classification (L_A)							
Static	0.40	0.39	0.38	0.39	0.50	0.50	0.50	0.50
Dynamic (1 violation)	0.47	0.49	0.50	0.49	0.50	0.49	0.50	0.50
Dynamic (2 violations)	0.47	0.50	0.50	0.49	0.50	0.50	0.50	0.50
Total	0.45	0.46	0.46	0.46	0.50	0.50	0.50	0.50

Table 3.15: Block O2 | Model: CNN aggregated

	Visible				Occluded			
Type of scene	1 obj.	2 obj.	3 obj.	Total	1 obj.	2 obj.	3 obj.	Total
	Relative classification (L_R)							
Static	0.00	0.00	0.01	0.00	0.50	0.47	0.52	0.50
Dynamic (1 violation)	0.13	0.22	0.25	0.20	0.51	0.50	0.55	0.52
Dynamic (2 violations)	0.11	0.10	0.17	0.13	0.56	0.49	0.53	0.53
Total	0.08	0.11	0.14	0.11	0.52	0.49	0.54	0.52
	Absolute classification (L_A)							
Static	0.34	0.41	0.40	0.38	0.50	0.50	0.50	0.50
Dynamic (1 violation)	0.43	0.45	0.46	0.45	0.50	0.50	0.50	0.50
Dynamic (2 violations)	0.43	0.44	0.46	0.45	0.50	0.50	0.50	0.50
Total	0.40	0.43	0.44	0.43	0.50	0.50	0.50	0.50

Table 3.16: Block O3 | Model: CNN (short-term prediction task)

	Visible				Occluded			
Type of scene	1 obj.	2 obj.	3 obj.	Total	1 obj.	2 obj.	3 obj.	Total
	Relative classification (L_R)							
Static	0.00	0.00	0.00	0.00	0.48	0.43	0.46	0.46
Dynamic (1 violation)	0.34	0.38	0.46	0.39	0.47	0.48	0.50	0.49
Dynamic (2 violations)	0.47	0.45	0.45	0.45	0.52	0.51	0.53	0.52
Total	0.27	0.27	0.30	0.28	0.49	0.47	0.49	0.49
	Absolute classification (L_A)							
Static	0.22	0.21	0.23	0.22	0.50	0.49	0.50	0.50
Dynamic (1 violation)	0.49	0.49	0.49	0.49	0.50	0.50	0.50	0.50
Dynamic (2 violations)	0.49	0.49	0.50	0.49	0.50	0.50	0.50	0.50
Total	0.40	0.40	0.41	0.40	0.50	0.50	0.50	0.50

Table 3.17: Block O3 | Model: GAN (short-term prediction task)

	Visible				Occluded			
Type of scene	1 obj.	2 obj.	3 obj.	Total	1 obj.	2 obj.	3 obj.	Total
	Relative classification (L_R)							
Static	0.00	0.00	0.00	0.00	0.47	0.43	0.37	0.43
Dynamic (1 violation)	0.31	0.45	0.43	0.40	0.50	0.47	0.54	0.50
Dynamic (2 violations)	0.34	0.42	0.43	0.40	0.48	0.52	0.54	0.51
Total	0.22	0.29	0.29	0.26	0.48	0.48	0.48	0.48
	Absolute classification (L_A)							
Static	0.29	0.33	0.30	0.31	0.50	0.49	0.50	0.50
Dynamic (1 violation)	0.46	0.49	0.49	0.48	0.50	0.50	0.51	0.50
Dynamic (2 violations)	0.44	0.47	0.47	0.46	0.50	0.50	0.50	0.50
Total	0.40	0.43	0.42	0.42	0.50	0.50	0.50	0.50

Table 3.18: Block O3 | Model: CNN (long-term prediction task)

	Visible				Occluded			
Type of scene	1 obj.	2 obj.	3 obj.	Total	1 obj.	2 obj.	3 obj.	Total
	Relative classification (L_R)							
Static	0.02	0.00	0.00	0.01	0.48	0.49	0.47	0.48
Dynamic (1 violation)	0.45	0.52	0.43	0.47	0.54	0.48	0.53	0.52
Dynamic (2 violations)	0.56	0.45	0.44	0.48	0.51	0.59	0.52	0.54
Total	0.35	0.32	0.29	0.32	0.51	0.52	0.51	0.51
	Absolute classification (L_A)							
Static	0.35	0.36	0.40	0.37	0.50	0.50	0.50	0.50
Dynamic (1 violation)	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
Dynamic (2 violations)	0.51	0.50	0.50	0.50	0.50	0.50	0.50	0.50
Total	0.45	0.45	0.47	0.46	0.50	0.50	0.50	0.50

Table 3.19: Block O3 | Model: GAN (long-term prediction task)

	Visible				Occluded			
Type of scene	1 obj.	2 obj.	3 obj.	Total	1 obj.	2 obj.	3 obj.	Total
	Relative classification (L_R)							
Static	0.01	0.00	0.00	0.00	0.53	0.53	0.59	0.55
Dynamic (1 violation)	0.53	0.50	0.60	0.54	0.55	0.55	0.48	0.53
Dynamic (2 violations)	0.42	0.51	0.54	0.49	0.43	0.52	0.52	0.49
Total	0.32	0.34	0.38	0.34	0.50	0.53	0.53	0.52
	Absolute classification (L_A)							
Static	0.29	0.33	0.35	0.32	0.50	0.50	0.51	0.50
Dynamic (1 violation)	0.50	0.49	0.52	0.50	0.50	0.51	0.50	0.50
Dynamic (2 violations)	0.50	0.49	0.50	0.50	0.50	0.50	0.50	0.50
Total	0.43	0.44	0.46	0.44	0.50	0.50	0.50	0.50

Table 3.20: Block O3 | Model: CNN aggregated

	Visible				Occluded			
Type of scene	1 obj.	2 obj.	3 obj.	Total	1 obj.	2 obj.	3 obj.	Total
	Relative classification (L_R)							
Static	0.02	0.00	0.00	0.01	0.48	0.49	0.47	0.48
Dynamic (1 violation)	0.45	0.52	0.43	0.47	0.54	0.48	0.53	0.52
Dynamic (2 violations)	0.56	0.45	0.44	0.48	0.51	0.59	0.52	0.54
Total	0.35	0.32	0.29	0.32	0.51	0.52	0.51	0.51
	Absolute classification (L_A)							
Static	0.35	0.36	0.40	0.37	0.50	0.50	0.50	0.50
Dynamic (1 violation)	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
Dynamic (2 violations)	0.51	0.50	0.50	0.50	0.50	0.50	0.50	0.50
Total	0.45	0.45	0.47	0.46	0.50	0.50	0.50	0.50

3.7.2 Human results (detailed)

Table 3.21: Block O1 | Human evaluation | Relative classification (L_R)

	Visible				Occluded			
Type of scene	1 obj.	2 obj.	3 obj.	Total	1 obj.	2 obj.	3 obj.	Total
Static	0.01	0.06	0.00	0.02	0.12	0.22	0.20	0.18
Dynamic (1 violation)	0.04	0.19	0.18	0.14	0.06	0.12	0.17	0.12
Dynamic (2 violations)	0.04	0.25	0.09	0.13	0.26	0.10	0.13	0.16
Total	0.03	0.17	0.09	0.10	0.15	0.15	0.17	0.15

Table 3.22: Block O2 | Human evaluation | Relative classification (L_R)

	Visible				Occluded			
Type of scene	1 obj.	2 obj.	3 obj.	Total	1 obj.	2 obj.	3 obj.	Total
Static	0.00	0.03	0.02	0.02	0.14	0.18	0.17	0.16
Dynamic (1 violation)	0.16	0.04	0.22	0.14	0.12	0.23	0.09	0.15
Dynamic (2 violations)	0.17	0.25	0.33	0.25	0.20	0.23	0.18	0.20
Total	0.11	0.11	0.19	0.14	0.15	0.21	0.15	0.17

Table 3.23: Block O3 | Human evaluation | Relative classification (L_R)

	Visible				Occluded			
Type of scene	1 obj.	2 obj.	3 obj.	Total	1 obj.	2 obj.	3 obj.	Total
Static	0.23	0.10	0.24	0.19	0.32	0.17	0.40	0.30
Dynamic (1 violation)	0.24	0.29	0.32	0.28	0.44	0.60	0.50	0.51
Dynamic (2 violations)	0.06	0.21	0.20	0.16	0.38	0.57	0.44	0.46
Total	0.18	0.20	0.25	0.21	0.38	0.45	0.45	0.42

Detailed mask predictor

Table 3.24: Mask predictor (9747011 parameters). BN stands for batch-normalization.

Input frame 3 x 64 x 64
7 first layers of resnet-18 (pretrained, frozen weights)
Reshape 1 x 8192
FC 8192 $\rightarrow$ 128
FC 128 $\rightarrow$ 8192
Reshape 128 x 8 x 8
UpSamplingNearest(2), 3 x 3 Conv. 128 - 1 str., BN, ReLU
UpSamplingNearest(2), 3 x 3 Conv. 64 - 1 str., BN, ReLU
UpSamplingNearest(2), 3 x 3 Conv. 3 - 1 str., BN, ReLU
3 sigmoid Target mask

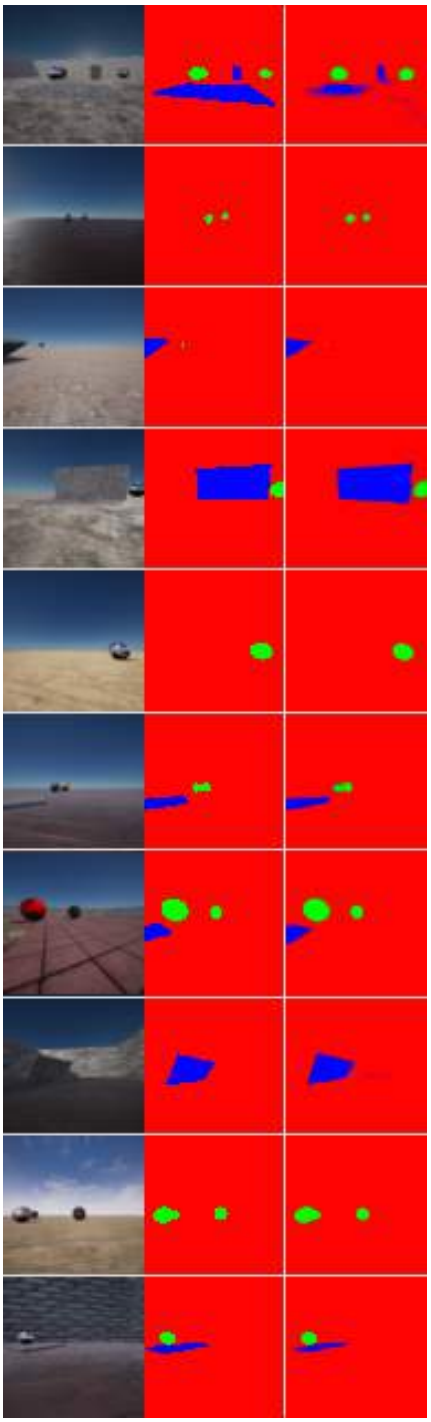


Figure 3.7: Output examples of our semantic mask predictor. From left to right: input image, ground truth semantic mask, predicted semantic mask.

Chapter 4

Occlusion resistant learning of intuitive physics from videos

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. (2021). 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.

This work was led in collaboration with Josef Sivic, Ivan Laptev and Emmanuel Dupoux.

4.1 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 and Carey (2012). This includes the notions of gravity Carey (2009), continuity of trajectories Spelke et al. (1995), collisions Saxe and Carey (2006), etc. Object permanence, the fact that an object continues to exist when it is occluded, Kellman and 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 and 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 is 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 prob-

abilistic 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 IntPhys benchmark Riochet et al. (2021). We show better performance compared to Riochet et al. (2021); 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. (2017), a first step towards using such models on real videos.

4.2 Related work

Forward modeling in videos. Forward modeling in video 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 has to generate frames in future time steps. To succeed in this task, such models need to predict object movements, suggesting that they need to learn physical regularities from video. However, models for end-to-end future frame prediction tend to perform poorly on long-term prediction tasks (say more 5-8 frames Lan et al. (2014); Mathieu et al. (2015); Finn et al. (2016)), failing to preserve object properties and generating blurry

outputs. This suggests that models for intuitive physics may require a more structured representation of objects and their interactions.

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 a off-the-shelf physics engine on both synthetic and real data Battaglia et al. (2013); Wu et al. (2017b); Smith 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); Fraccaro et al. (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, Fraccaro et al. (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. (2019), 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), 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 properties of objects. 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. (2016, 2017); 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 Greff et al. (2019).

4.3 Occlusion resistant intuitive physics

This section describes our model for occlusion resistant learning of intuitive physics. In section 4.3.1 we present an 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 4.3.2), and the recurrent interaction network that predicts the scene state evolution over time (section 4.3.3). Finally, in section 4.3.4 we describe how these two components are used jointly to decode an entire video clip.

4.3.1 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: $\hat{S} = \text{argmax}_S\, P(S|F,\theta)$. 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 Neufeld (1999). The event decoding problem then becomes:

$$\hat{S} = \underset{S}{\text{argmax}}\, P(F|S,\theta)P(S|\theta). \tag{4.1}$$

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:

$$\hat{\theta} = \underset{\theta}{\text{argmax}}\, P(F|\theta). \tag{4.2}$$

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. (2017)), 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 argmax in eq. (4.1) 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. (4.1)) 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:

$$\begin{aligned} L_{\text{render}}(S, F) &= \sum_{t=1}^{T} L_{\text{mask}}(Renderer(s_t), F), \\ L_{\text{physics}}(S) &= \sum_{t=1}^{T-1} \|s_{t+1} - RecIntNet(s_t)\|^2, \\ L_{\text{total}}(S, F) &= \lambda L_{\text{render}}(S, F) + (1-\lambda) L_{\text{physics}}(S). \end{aligned} \tag{4.3}$$

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 4.4.1). 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.

4.3.2 The Compositional Renderer ($Renderer$)

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 4.2). 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.

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.).

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 up-sample (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 $i, j \leqslant 128$ and object k the following weight:

$$c_{i,j}^k = \frac{e^{\lambda \hat{D}_{i,j}^k}}{\sum_{q=1}^N e^{\lambda \hat{D}_{i,j}^q}}, k = 1..N, \tag{4.4}$$

where λ is a parameter learned by the model. The final masks and depth maps are computed as a weighted combination of masks $\hat{M}_{i,j}^k$ and depth maps $\hat{D}_{i,j}^k$ for individual objects k: $\hat{M}_{i,j} = \sum_{k=1}^N c_{i,j}^k \hat{M}_{i,j}^k$, $\hat{D}_{i,j} = \sum_{k=1}^N c_{i,j}^k \hat{D}_{i,j}^k$, where i, j are output pixel coordinates $\forall i, j \leqslant 128$ and $c_{i,j}^k$ the weights given by (5.8). 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 (5.8) 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).

4.3.3 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:

$$[p_1, v_1, c] = [p_0 + \delta t v_0 + \frac{\delta t^2}{2}\mathbf{d_v}, v_0 + \mathbf{d_v}, c], \tag{4.5}$$

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_x^2, σ_y^2, σ_d^2 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:

$$\mathcal{L}\big((\hat{p_1}, \hat{\tau_1}), p_1\big) = \frac{(\hat{p_1} - p_1)^2}{\exp \hat{\tau_1}} + \hat{\tau_1}, \tag{4.6}$$

where $\hat{\tau_1} = \ln \hat{\sigma_1^2}$ 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.

4.3.4 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, as well as it's plausibility score (inverse of the total loss). The details of this algorithm are given in the supplementary material.

4.4 Experiments

In this section we present two sets of experiments evaluating the proposed model. The first set of experiments (section 4.4.1) is on the IntPhys benchmark that is becoming the de facto standard for evaluating models of intuitive physics[1] Riochet

[1] www.intphys.com

et al. (2021), and is currently used as evaluation in the DARPA Machine Common Sense program. The second set of experiments (section 4.4.2) 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.

4.4.1 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. (2021); 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 4.1 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. (2021) 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. (2021), 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 4.1. 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. (2021), 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.

	Block O1			Block O2			Block O3		
	V	O	Total	V	O	Total	V	O	Total
Ours	**0.05**	**0.19**	**0.12**	**0.11**	**0.31**	**0.21**	**0.26**	**0.47**	**0.37**
Riochet et al. (2021)	0.07	0.52	0.29	**0.11**	0.52	0.31	0.32	0.51	0.41
Smith et al. (2019)	-	-	0.27	-	-	-	-	-	-

Table 4.1: **Results on the IntPhys benchmark.** Relative classification error of our model compared to Riochet et al. (2021) and Smith et al. (2019), demonstrating large benefits of our method in scenes with significant occlusions ("Occluded"). V stands for Visible and O for Occluded. Lower is better.

4.4.2 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 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 4.3. We first note that our model (e. RecIntNet) significantly outperforms the linear baseline (a.), which is computed as an extrapolation of the

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

	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 4.2: **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.

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.

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 4.3). This MLP contains four hidden layers of size 180 and is trained the same way as $RecIntNet$, modeling acceleration as described in equation 5.4.1. 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. 5.4.1). This is similar to the model described in Janner et al. (2019), where object representation is not decomposed into position / velocity / intrinsic properties, but is rather a (unstructured) 256-dimensional vector. Results are reported in table 4.3 (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 5.4.1) 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 4.3 (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 the data are shown in figure 4.7. 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).

4.5 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 anonymous google drive (link). This work, along with recent improvement of object segmentation models Ren et al. (2017) 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.

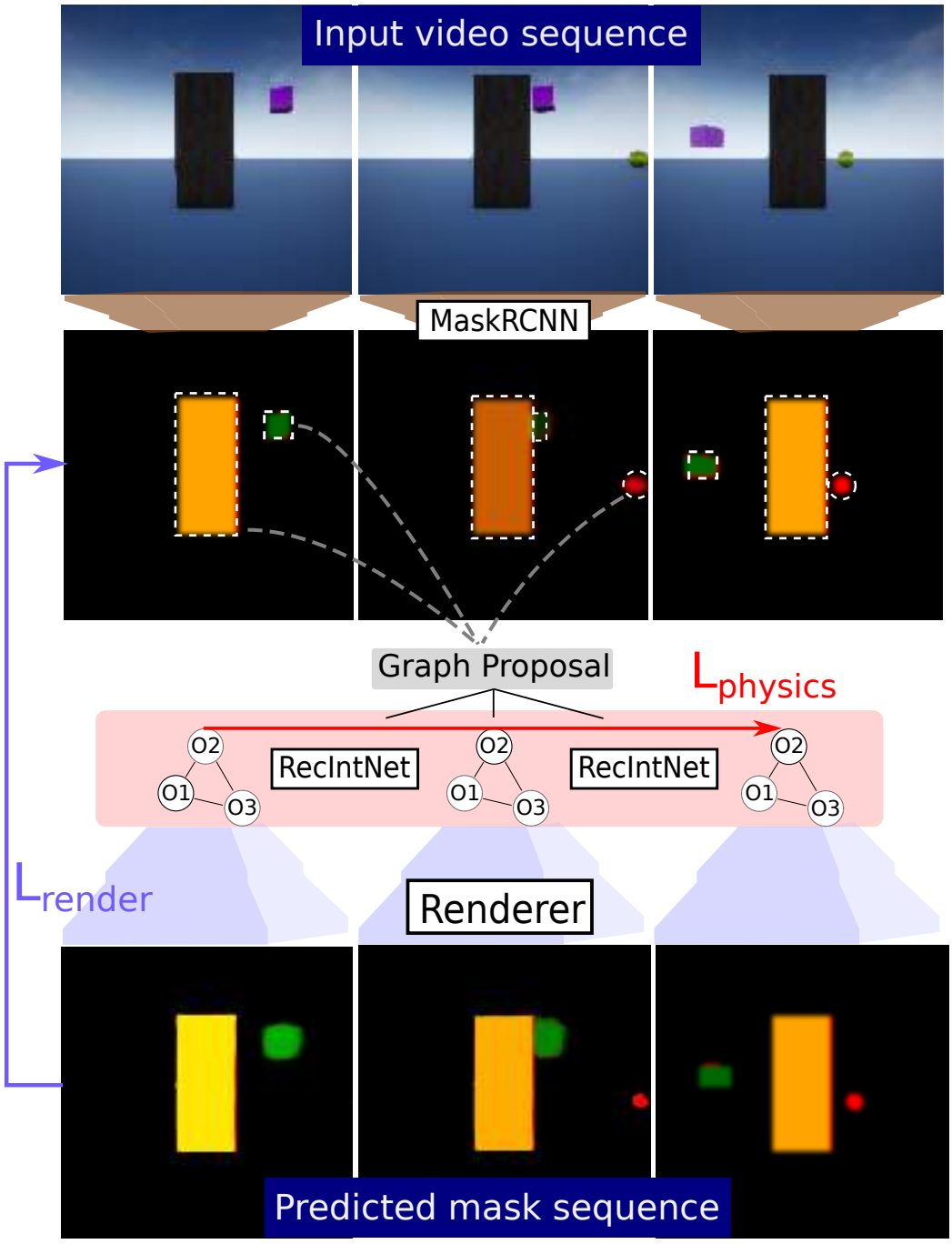


Figure 4.1: **Overview of our occlusion resistant intuitive physics model.** A pre-trained object detector (MaskRCNN) returns object detections and masks (top). A *graph proposal* matching links object proposals through time: from a pair of frames the Recurrent Interaction Network ($RecIntNet$) predicts 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.

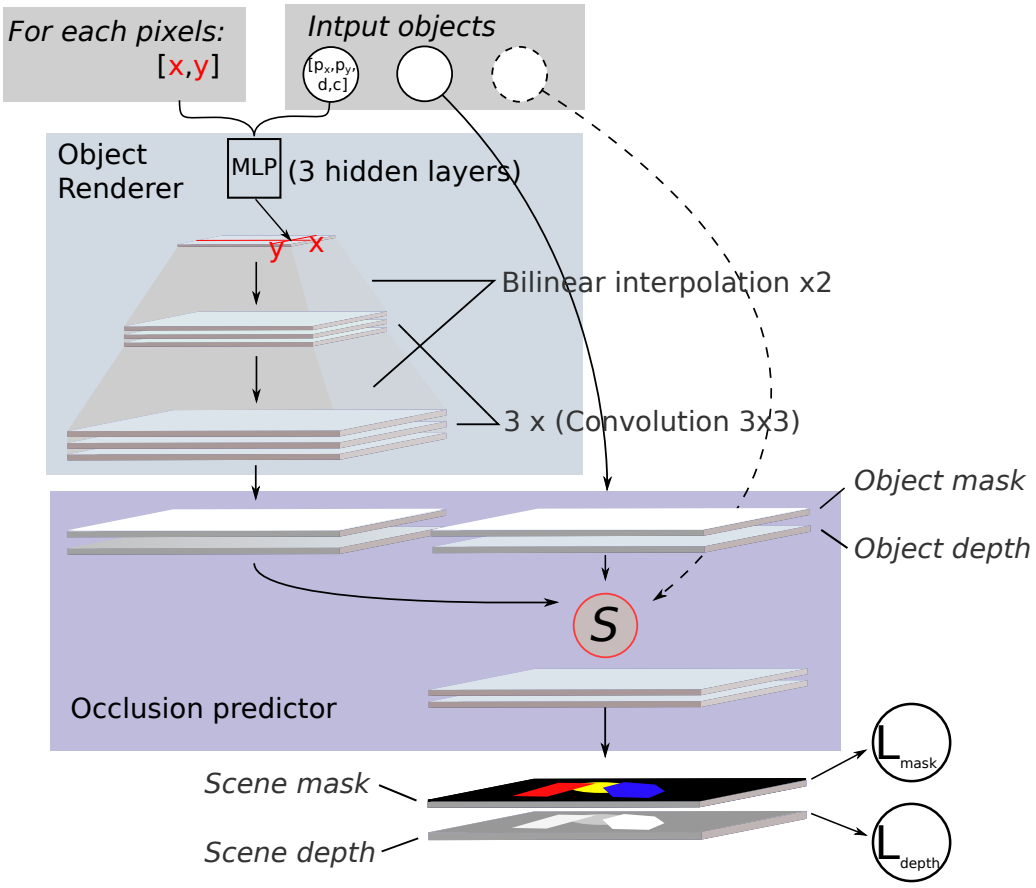


Figure 4.2: **Compositional Rendering Network** (*Renderer*).

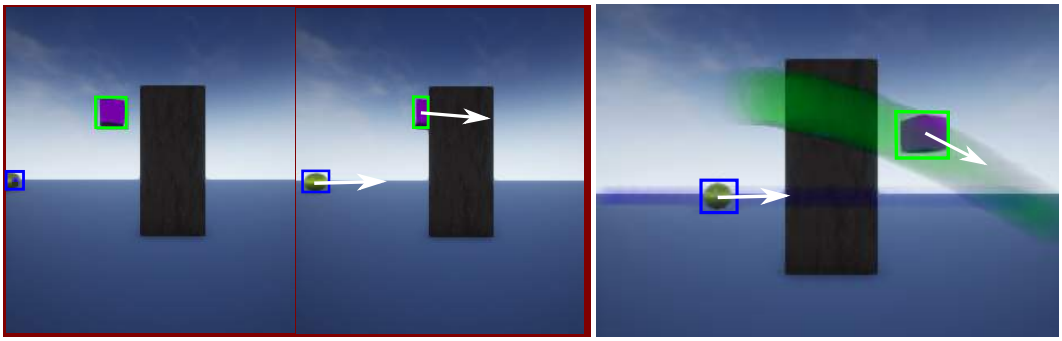


Figure 4.3: **Illustration of event decoding in the videos of the IntPhys dataset.** A pre-trained object detector returns object proposals in the video (bounding boxes). An initial match is made across two seed neighbouring frames, also estimating object velocity (left, white arrows). The dynamic model (RecIntNet) predicts object positions and velocities in future frames, enabling the match of objects despite significant occlusions (right, bounding box colors and highlights).

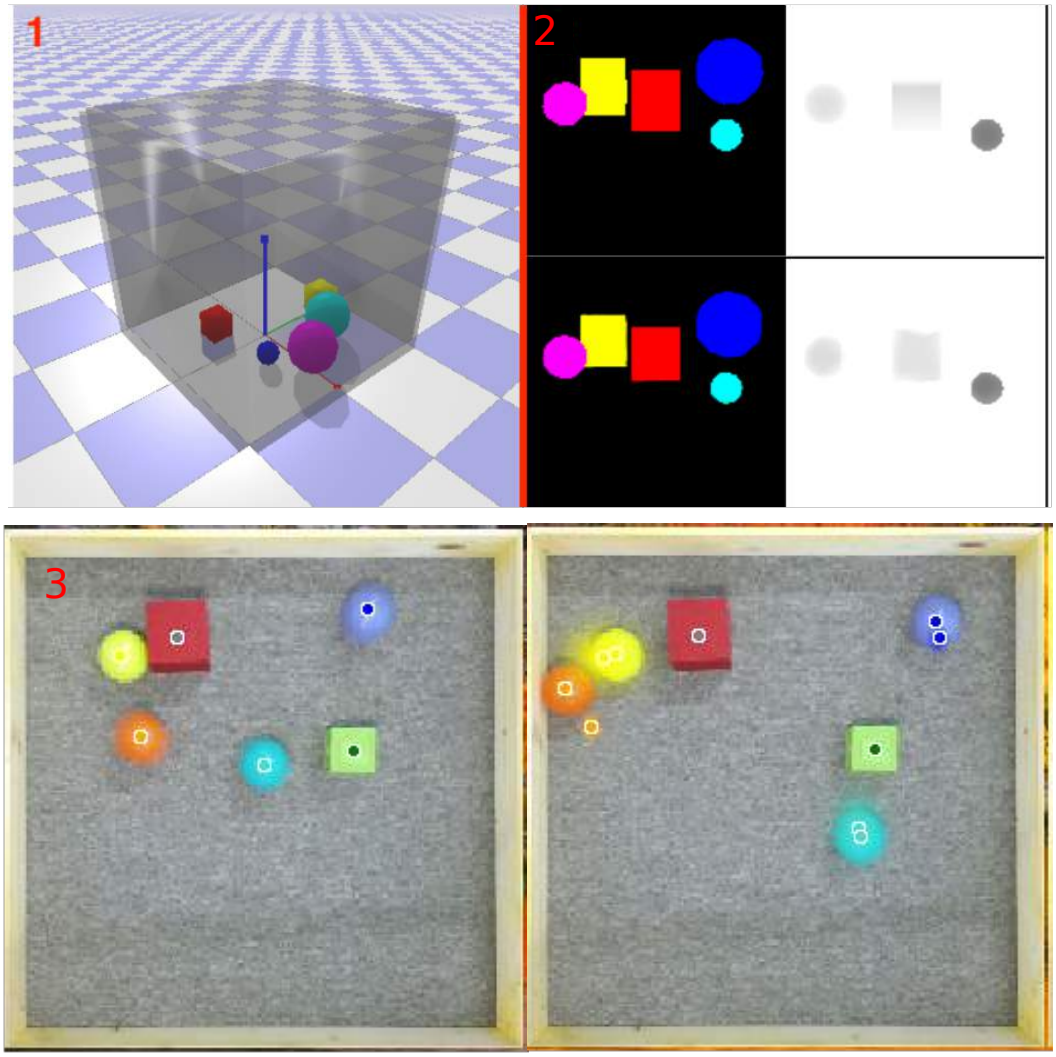


Figure 4.4: **Images from the Future Prediction experiment** 1: An overview of the pybullet scene. 2: Sample video frames (instance mask + depth field) from our datasets (top) together with predictions obtained by our model (bottom), taken from from the tilted 25° experiments. 3: example of prediction 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).

4.6 Appendix

This supplementary material: (i) describes the provided supplementary videos (section 4.6.1), (ii) provides additional training details (section 4.6.2), (iii) explains in more depth the *event decoding* procedure defined in section 3.4 in the main paper (section 4.6.2) (iv) gives details of the datasets used in the subsection 4.2 (section 4.6.2), (v) provides additional ablation studies and comparisons with baselines (sections 4.6.2, 4.6.3, 4.6.4, 4.6.5).

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

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.

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.

Real videos

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

4.6.2 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

$$L_{\text{mask}}(\hat{M}, M) = \sum_{i \leqslant h, j \leqslant w} \sum_{n \leqslant N} \mathbf{1}(M_{i,j} = n) log(\hat{M}_{i,j,n}), \tag{4.7}$$

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

$$L_{\text{depth}}(\hat{D}, D) = \sum_{i \leqslant h, j \leqslant w} (\hat{D}_{i,j} - D_{i,j})^2, \tag{4.8}$$

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.

Event Decoding

The detailed outline of the event decoding procedure described in section 3.4 of the main paper is given in Algorithm 1. Two example figures (Figure 4.5 & 4.6) gives an intuition behind the *render* and *physics* losses.

Datasets

To validate our model, we use pybullet[3] 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

[3] `www.pypi.org/project/pybullet`

Algorithm 2: 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:

$d_{t=1..T} = \text{Detection}(f_t)$;
$n_t \leftarrow (\#\{d_t\}, \text{mean}_t size(d_t))$;
$t^* \leftarrow \text{argmax}_t(n_t + n_{t+1})$;
`//`(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.`
$\text{Loss}_{\text{physics}}(s) \leftarrow \sum_{t=1}^{T} \|\hat{s}_{t+1} - s_{t+1}\|^2$;
$\text{Loss}_{\text{visual}}(s) \leftarrow \sum_{t=1}^{T} \texttt{NLL}(\texttt{Rend}(s_t), m_t)$;
$\text{Loss}_{\text{plausibility}}(s) \leftarrow \lambda \text{Loss}_{\text{physics}}(s) + (1-\lambda)\text{Loss}_{\text{visual}}(s)$;
(Estimated states, plausibility loss) $\leftarrow \texttt{SGD}_s(\text{Loss}_{\text{plausibility}}(s))$;
`//with` $lr = 1e-3$ `and` $n_{\texttt{steps}} = 1000$;

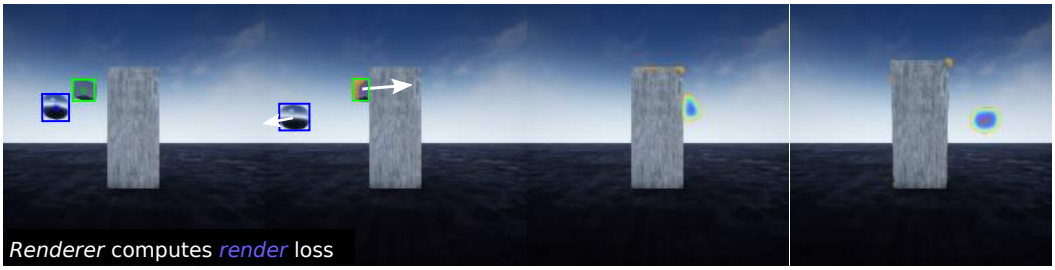


Figure 4.5: **Video example from the IntPhys benchmark.** Four frames from a video in block O1, with superimposed heatmaps. Heatmaps (colored blobs) correspond to the difference, per pixel, between the predicted and the observed object mask. In these video, a cube moves from left to right but disappears 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.

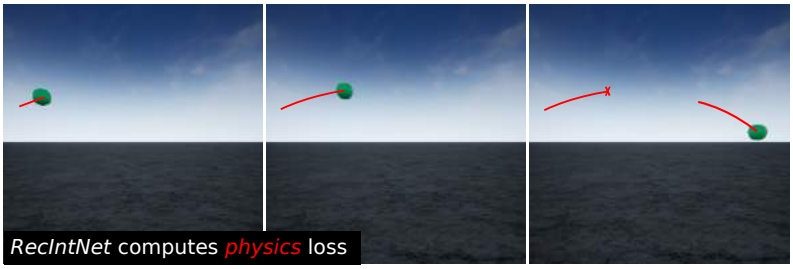


Figure 4.6: **Video example from the IntPhys benchmark.** Three frames from a video in block O2, where an object "jumps" from one place to another. The graph proposal phase returns the right trajectory of the object but the Recurrent Interaction Network returns a high *physics* loss at the moment of the jump, because the observed position is far from the predicted one.

(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

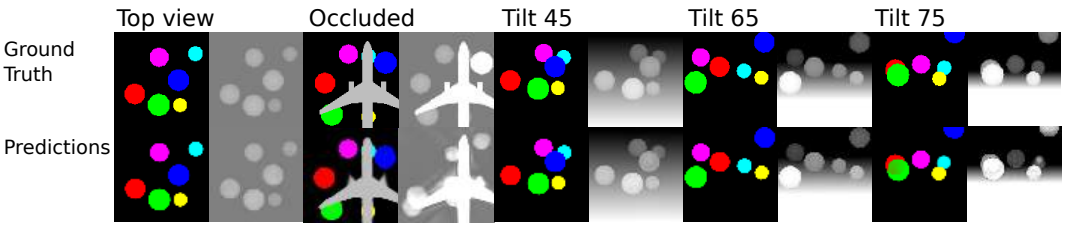


Figure 4.7: Sample video frames (instance mask + depth field) from our datasets (top) together with predictions obtained by our model (bottom). Taken from the top-view, occluded and tilted experiments. **Please see additional video results in the anonymous google drive `https://drive.google.com/open?id=1Qc8flIAxUGzfRfeFyyUEGXe6J5AUGUjE`.**

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 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.

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 4.3: 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.

4.6.3 Roll-out results

We evaluate our model on *object following*, applying an online variant of the scene decoding procedure detailed in 4.6.2. 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 4.4 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 4.8, 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.

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 4.4: 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.

4.6.4 Experiment with real videos

We construct a dataset of 22 real videos, containing a variable number of colored balls and blocks in motion. Videos are recorded with a Microsoft Kinect2 device,

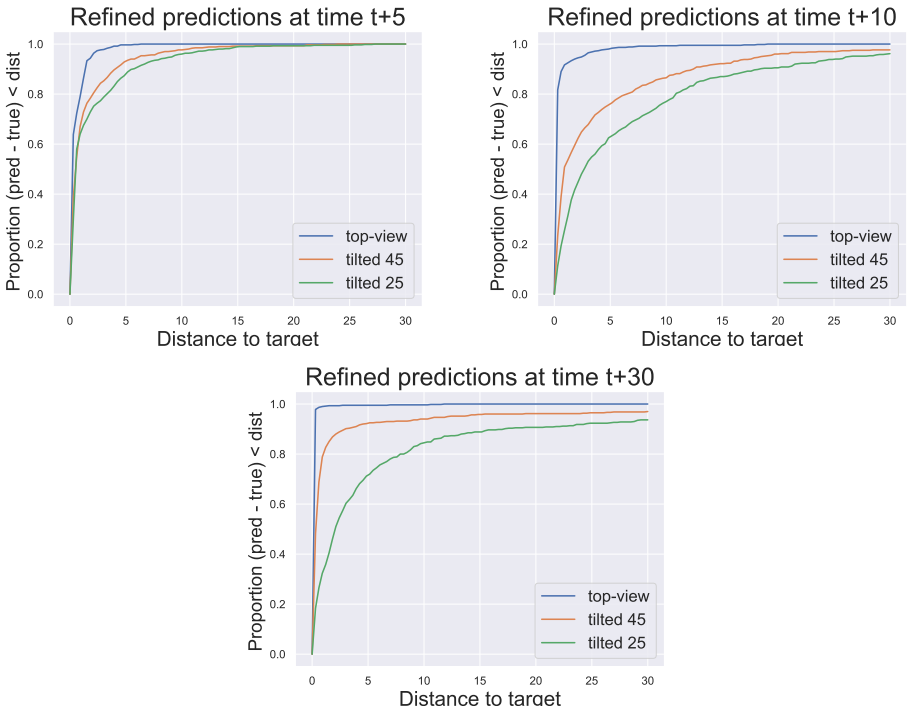


Figure 4.8: Proportion of correctly followed objects (y-axis) as a function of the distance error in pixels (x-axis) for our approach using online event decoding. The different plots correspond to rollout lengths of 5 (left), 10 (middle) and 30 (right) frames. Different curves correspond to different 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".**

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

Table 4.5: **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.

Figure 4.9: 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).

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 4.7). 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 4.5 and clearly demonstrate that out approach outperforms the linear and MLP baselines and makes reasonable predictions on real videos.

	Top view	Top view+ occlusion	45° tilt	25° tilt	15° tilt
CNN autoencoder Riochet et al. (2021)	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

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).

4.6.5 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 4.6.2. Here, we compare our model to a CNN autoencoder Riochet et al. (2021), 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. (2021).

Chapter 5

Multi-Representational Future Forecasting

Abstract

Understanding the dynamics of an environment from a visual input is an essential component of reasoning. Improving intuitive physics skills of machine models is therefore important, however models are often validated using toy datasets, with static cameras. To ease the complex task of future prediction in a real-world context, we combine in this work the use of different representations: objects bounding boxes, keypoints, instances, and background masks. We study predicting each representation separately as well as conditioning complex representations on simpler ones for a more accurate prediction. Possible influences between objects are modeled via an interaction network. We first utilize synthetic labels for learning to predict ideal state representations, and investigate a domain transition using real data with labels obtained by an automatic detection system. Our interaction modeling followed by geometric projection allows us to outperform the state-of-the-art in future instance segmentation, with more than 15% of relative improvement. Our prediction and data generation codes will be made publicly available.

This work was led in collaboration with Mohamed Elfeki, Natalia Neverova, Emmanuel Dupoux and Camille Couprie.

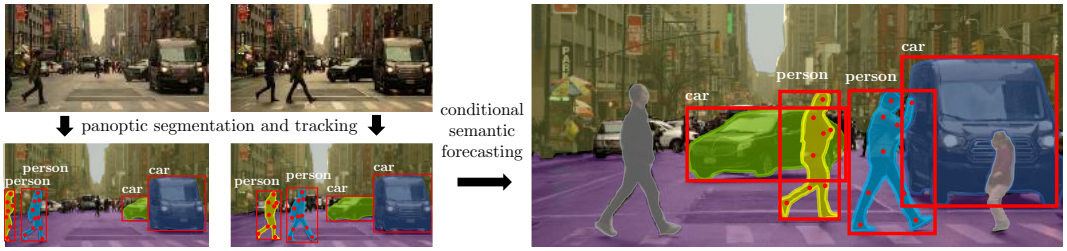


Figure 5.1: Can we use a multi-modal state representation to reduce the accumulating entropy within future prediction? We investigate different modalities and their sequential conditioning to perform long-range predictions. Additionally, we model objects relationships using an interaction network and perform dense instance masks predictions through a renderer.

5.1 Introduction

Providing intelligent agents with the ability to predict and anticipate has been a topic of extensive research for a long time Kitani et al. (2012)Mathieu et al. (2015)Alahi et al. (2016)Zhang et al. (2017), and can be directly applicable in the domains of robotics Koppula and Saxena (2015), autonomous driving McAllister et al. (2017); Henaff et al. (2019), and building intelligent assistants, to name a few. Meanwhile, the natural order of events presented in dynamic visual sequence provides a source of unlimited natural supervision for learning powerful spatio-temporal representations that are transferable to other downstream problems, such as action recognition Lee et al. (2017); Xu et al. (2019a).

The task of forecasting the visual future encompasses a variety of sub-tasks, from simply extrapolating trajectories of objects in the scene, to a full-fledged generation of high resolution frames for extending observed video sequences. The latter problem of video prediction directly in the pixel space is known to be notoriously hard and requires scaling the training process to massive amounts of data and compute to achieve a modest degree of realism even over a short prediction range Weissenborn et al. (2019); Clark et al. (2019). In addition, there exists no appropriate evaluation metric for identifying and interpreting shortcomings of such systems, as well as quantifying their reasoning capabilities. For this reason, many works have attempted to constrain the task by introducing inductive biases, such as decomposing the scene

into a set of entities, or agents, to model their respective trajectories and pairwise interactions Ye et al. (2019); Baradel et al. (2019). Another popular strategy is shifting the reconstruction from RGB space to *semantic spaces*, such as semantic segmentation Luc et al. (2017), object masks Luc et al. (2018); Ye et al. (2019) or human keypoints Walker et al. (2017); Kim et al. (2019). This is based on the intuition that a model performing reconstruction of object dynamics should be invariant to object appearance, which is thus irrelevant to the prediction task. At the same time, recent works in generative modeling have indicated that having a semantic representation of a scene (such as parsing Wang et al. (2018), scene graph Ashual and Wolf (2019) or object skeleton Walker et al. (2017); Kim et al. (2019)) in a new frame is indeed sufficient to perform realistic texture transfer from past observations.

A crucial aspect of future prediction is the modeling of object interactions. In intuitive physics, the Interaction networks of Battaglia et al. (2016) were introduced to perform such predictions. The work of Riochet et al. (2020b) goes one step further and applies it to higher dimension simulated images of moving balls. In the context of real world datasets, graph-based modelings are for now limited to trajectory forecasting Alahi et al. (2016); Ma et al. (2019).

The goal of the present study is to address the problem of *semantic video prediction* in a systematic way, combining different representations, modeling their relationships and building on the progress in the field so far. We consider this task in a setting of *semantic multi-modality*, by considering a set of semantic representations (shown in Fig. 5.1) ordered by their expressive power (from object locations to keypoints and masks). We also model objects interactions and render them spatially. We start by conducting the experiments by adapting the synthetic CARLA environment for autonomous driving Dosovitskiy et al. (2017) to generate a large scale dataset for multi-modal visual forecasting. Then we generalize the framework's performance on the real-life Cityscapes dataset Cordts et al. (2016) using a state-of-the-art semantic content extractor, here Mask-RCNN He et al. (2017). By gathering all major pieces: conditioning complex modality prediction on simpler ones, decoupling objects' relative motion from background, and generalizing framework's performance on real-data in a self-supervised setting, we aim at taking a first stride towards a precise multi-modal environment dynamic modeling.

We present a forecasting model that decomposes the visual input into foreground

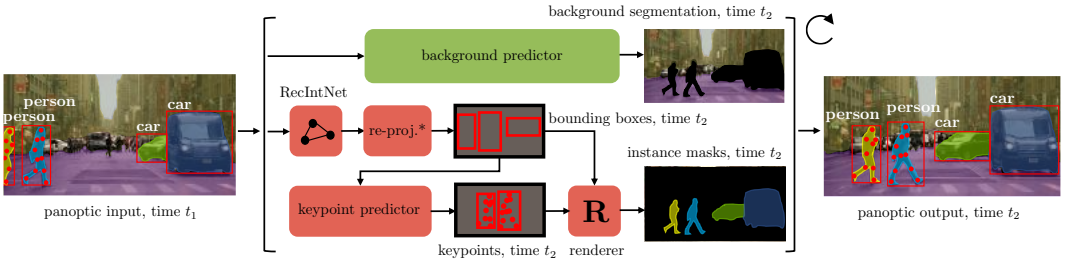


Figure 5.2: Our multi-representational prediction pipeline. A recursive interaction network is trained to predict future bounding box locations, that are then fed to an LSTM along with past keypoints to predict the future keypoints. Then our re-projection module corrects these prediction using a geometric projection when camera information is available. Corrected bounding boxes and keypoints are fed to our renderer, along with previous masks to generates future instance masks. *: optional.

instance representations, object bounding boxes, keypoints, and masks. We demonstrate the positive impact of conditioning predictions in different semantic spaces on each other in increasing order of complexity of representations and expressive power (from bounding boxes to keypoints and masks).

Our method effectively decouples egomotion from objects' motion in the scene by implementing a dedicated correction module using perspective projection of camera information. This is necessary for most robotic applications, where both the camera and the environment may be in motion.

5.2 Related work

Most prior literature on video prediction focuses on a single modality at a time, including these that we consider in this work. Below we briefly review the existing works on visual semantic forecasting.

Bounding box forecasting. The recent work of Bhattacharyya et al. (2018) proposes a method to predict the boxes around pedestrians using a non deterministic loss and a LSTM on the ego-centric Cityscapes sequences. The framework of Yao et al. (2019) includes a multi-stream recurrent neural network (RNN) encoder-decoder model to predict bounding boxes for vehicles. Additionally, the authors use

dense optical flow to capture motion and appearance changes.

Keypoint forecasting. For their representative power, keypoint predictions have been used immensely for activity recognition Walker et al. (2017); Jain et al. (2016). Various methods have been introduced to predict the human-pose whether as action-agnostic such as Chiu et al. (2019), or learning the human dynamics such as Fragkiadaki et al. (2015b); Villegas et al. (2018). Nonetheless, most prior literature focuses on human-based keypoint extraction and prediction, which is not well-generalized to a generic scene that contains other foreground dynamic instances such as cars and bicycles. Unlike human pose estimation Toshev and Szegedy (2014); He et al. (2017), there is a lack of generic and applicable keypoint extraction methods for non-human objects. The handful of vehicle keypoint extraction may assume extra supervisory signals such as Wu et al. (2019a)'s work that also predicts six degrees of freedom in 3D assuming monocular RGB images.

Segmentation forecasting. Among the many representations in the descriptive spaces (i.e., non-RGB spaces), segmentation is the most complex. It has a pixel-to-pixel mapping of each object, and sometimes of the background as well, and is hence modeling position, size, orientation and appearance changes only without coloring. Luc et al. (2017) propose predicting future *semantic segmentations* without a clear distinction of instances using a CNN architecture. Going beyond basic CNN architectures used for segmentation forecasting, Nabavi et al. (2018) employ bidirectional -LSTMs to segment all instances together, and perform very near future forecasting (next single frame prediction). The works of Terwilliger et al. (2019); Saric et al. (2020) successively improve the state-of-the-art by jointly inferring future optical flow and using warping to predict future semantic segmentations. Finally, Qi et al. (2019) uses optical flow and depth estimation to infer 3D point clouds and improve future segmentations before frames predictions.

Future *instance segmentation* was introduced by Luc et al. (2018), where the convolutional features obtained by a Mask-RCNN backbone He et al. (2017) are forecasted. Sun et al. Sun et al. (2019) build on this by modifying the architecture to use convolutional LSTMs. While combining both instance and semantic forecasting ideas is suggested by Couprie et al. (2018), the instance segmentations are not learned

	RGB frames	Instance segm.	Semantic segm.	car & bi-cycle kp	human kp	depth	camera info	object tracks
Carla GT recording	✓	✓	✓	✓	✓	✓	✓	✓
Carla infered annotations		✓	✓		✓			✓
Cityscapes infered annotations	✓*	✓	✓			✓*	✓*	✓

Table 5.1: Summary of our dataset. *: provided by the Cityscapes dataset.

end-to-end, resulting in weak performance.

5.3 Semantically multi-modal datasets

Collecting long-sequences of data, with labels of different modalities, and in abundance, is an extremely hard task that might be rendered impractical. Only a limited number of in-the-wild datasets contain several ground-truth annotation streams recorded simultaneously, and those labels are offered in scarcity. In particular, there exist no standard benchmark offering ground truth information for object tracking, depth estimation, object detection, keypoint detection, instance segmentation and semantic segmentation. For instance, the Cityscapes dataset of Cordts et al. (2016) provides semantic and instance segmentation, center locations, but no keypoints, and sequences of at most 1.5 seconds contain single labeled frames. To cope with this, we gathered synthetic and pseudo ground truth real annotations as described in Table 5.1.

Synthetic data. We use the Carla self-driving engine Dosovitskiy et al. (2017) to generate 450 driving sequences of long range (120 frames per sequence, corresponding to two minutes) and accurate labels representing multiple semantic modalities for every frame. Carla is a video generation engine that simulates a virtually-infinite number of realistic scenarios of an autonomous car driving in urban and rural areas as well as highways. It offers simulation for multiple types of moving agents, including cars, trucks, bicycles, bikes, pedestrians, with several subcategories of each agent. Another crucial importance of using a synthetic engine is acquiring access to simultaneously recorded ground-truth annotations for different modalities, which is impractical and cost-ineffective to be done in real life. We modified the engine to produce: 2D relative locations, bounding boxes, keypoints for cars and pedestrians,

depth maps, semantic and instance segmentations. We also provide meta information about each instance such as object-id, which is useful for tracking. Further details about the data collection are reported in the Supplementary material.

Real-world data. On both Carla and Cityscapes sequences, we use pseudo detection labels predicted by a panoptic segmentation network from Kirillov et al. (2019) implemented within the Mask-RCNN framework He et al. (2017). Specifically, we use Detectron2 Wu et al. (2019b) to obtain tracking information (using a heuristic to match similar instances), bounding box detection, instance segmentation, and keypoints for pedestrians.

5.4 Models

An overview of our future prediction system is depicted in Figure 5.2. To predict better instance segmentation, we start predicting object's locations and sizes in the scene, through a bounding box forecasting model. Objects motion being strongly linked to inter-object interactions, we model their trajectories with a Recurrent Interaction Network (Battaglia et al. (2016); Riochet et al. (2020b)). Keypoints also provide useful information for instance segmentation, such as object orientation / deformation. Because they are often more complex to model (e.g., legs of walkers, bicycles), we help the model catching these regularities by conditioning to the object bounding box.

5.4.1 Modeling interactions for object bounding box forecasting

Taking inspiration from the work of Riochet et al. Riochet et al. (2020b), we model relationships between different objects via a Recurrent Interaction Network.

Interaction Network. An interaction network Battaglia et al. (2016) consists in a graph neural network where objects are nodes and their interactions are vertices. A four-layers Multi-Layer Perceptron (MLP), with hidden states of length 150 and `ReLU` activation units predicts the result of all interactions: for each object

pair, it takes as input the concatenation of both object states and returns a latent representation of the interaction, encoded as a vector of size 100. To predict object motion, we aggregate all interactions involving this object by summing their latent representations, and apply a second 4-layers MLP with hidden states of length 100 and `ReLU` activation units.

Object state. An object state consists in its bounding box position $\mathbf{p}_t = [x, y, w, h]_t$, velocity $\mathbf{v}_t$ and the predicted object category encoded as a one-hot vector $\mathbf{l}$. Like in Riochet et al. (2020b), 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 predict changes in velocity $\mathbf{d_v} = \mathbf{v}_t - \mathbf{v}_{t-1}$, reconstructing object state as follow:

$$[\mathbf{p}_1, \mathbf{v}_1, \mathbf{l}] = [\mathbf{p}_0 + \delta t \mathbf{v}_0 + \frac{\delta t^2}{2}\mathbf{d_v}, \mathbf{v}_0 + \mathbf{d_v}, \mathbf{l}], \tag{5.1}$$

where $\mathbf{p}_1$ and $\mathbf{v}_1$ are position and velocity of the object at time t_1, $\mathbf{p}_0$ and $\mathbf{v}_0$ are position and velocity at time t_0, and $\delta t = t_1 - t_0$ is the time step. Hence $\mathbf{d_v}$ can be seen as the *acceleration*, and $(\mathbf{v}_0 + \mathbf{d_v})$,$(\mathbf{p}_0 + \delta t \mathbf{v}_0 + \frac{\delta t^2}{2}\mathbf{d_v})$ as the first and second order Taylor approximations of velocity and position, respectively. The bounding box is augmented with the distance of the object to the camera: $\mathbf{p}_t = [x, y, w, h, d]_t$, to help the model catch object dynamics. This distance can be either the ground truth depth of the object (e.g. in Carla) or estimated as the median of the depth map in the object's instance map.

Correcting forecasting using perspective projection. Optionally, we correct bounding box predictions using ego-motion information. Using camera position and orientation during the observation sequence, we apply an *inverse perspective projection* to all objects, decoupling their trajectories from ego-motion. For inference, we can project back these objects in the scene, applying a *perspective projection* conditioned by the new (or predicted) state of the camera. Importantly, this inverse perspective projection does not require absolute camera position but camera displacements between each frame. For Cityscapes, we compute this information by

integrating `speed` and `yaw rate` along the sequence, which is available in the Cityscapes dataset.

We consider a sequence of m frames, in which $n-1$ are observed . For $t < n$, we observe:

- $\mathbf{c}^t$: camera location
- θ^t: camera orientation
- $\mathbf{r}^t_{\mathbf{c}^t,\theta^t} = (p_x, p_y, d)^t_{\mathbf{c}^t,\theta^t}$: *relative* position of the object in the frame (e.g., from object detection) for camera $(\mathbf{c}^t, \theta^t)$.

We detail in the Supplementary material how to define the transformation $\texttt{pproj}$ and inverse transformation $\texttt{proj}^{-1}$. We compute the following:

$$\mathbf{r}^t_{\mathbf{c}^0,\theta^0} = \texttt{pproj}_{\mathbf{c}^0,\theta^0}(\texttt{pproj}^{-1}_{\mathbf{c}^t,\theta^t}(\mathbf{r}^t_{\mathbf{c}^t,\theta^t})),\ \forall t < n, \tag{5.2}$$

which is the trajectory which would be observed if the camera was fixed at its initial position/rotation. Note that the resulting trajectory is in a inertial space. Consider a dynamic model Dyn (e.g., LSTM, Interaction Network, Fixed-Velocity baseline).

$$\hat{\mathbf{r}}^n_{\mathbf{c}^0,\theta^0} = Dyn(\mathbf{r}^{n-1}_{\mathbf{c}^0,\theta^0}). \tag{5.3}$$

We predict the future sequence from this observation:

$$\hat{\mathbf{r}}^{t+1}_{\mathbf{c}^0,\theta^0} = Dyn(\hat{\mathbf{r}}^t_{\mathbf{c}^0,\theta^0}), \forall n \leqslant t < N-1. \tag{5.4}$$

We project back this prediction w.r.t. the actual position of the camera.

$$\hat{\mathbf{r}}^t_{\mathbf{c}^t,\theta^t} = \texttt{pproj}_{\mathbf{c}^t,\theta^t}(\texttt{pproj}^{-1}_{\mathbf{c}^0,\theta^0}(\hat{\mathbf{r}}^t_{\mathbf{c}^0,\theta^0})),\ \forall n \leqslant t < m. \tag{5.5}$$

Finally, we have a predicted sequence $\hat{\mathbf{r}}^t_{\mathbf{c}^t,\theta^t}, n \leqslant t < m$ which we can compare with the ground truth $\mathbf{r}^t_{\mathbf{c}^t,\theta^t}, n \leqslant t < m$. We show on Figure 5.3 predictions with and without correction for egomotion.

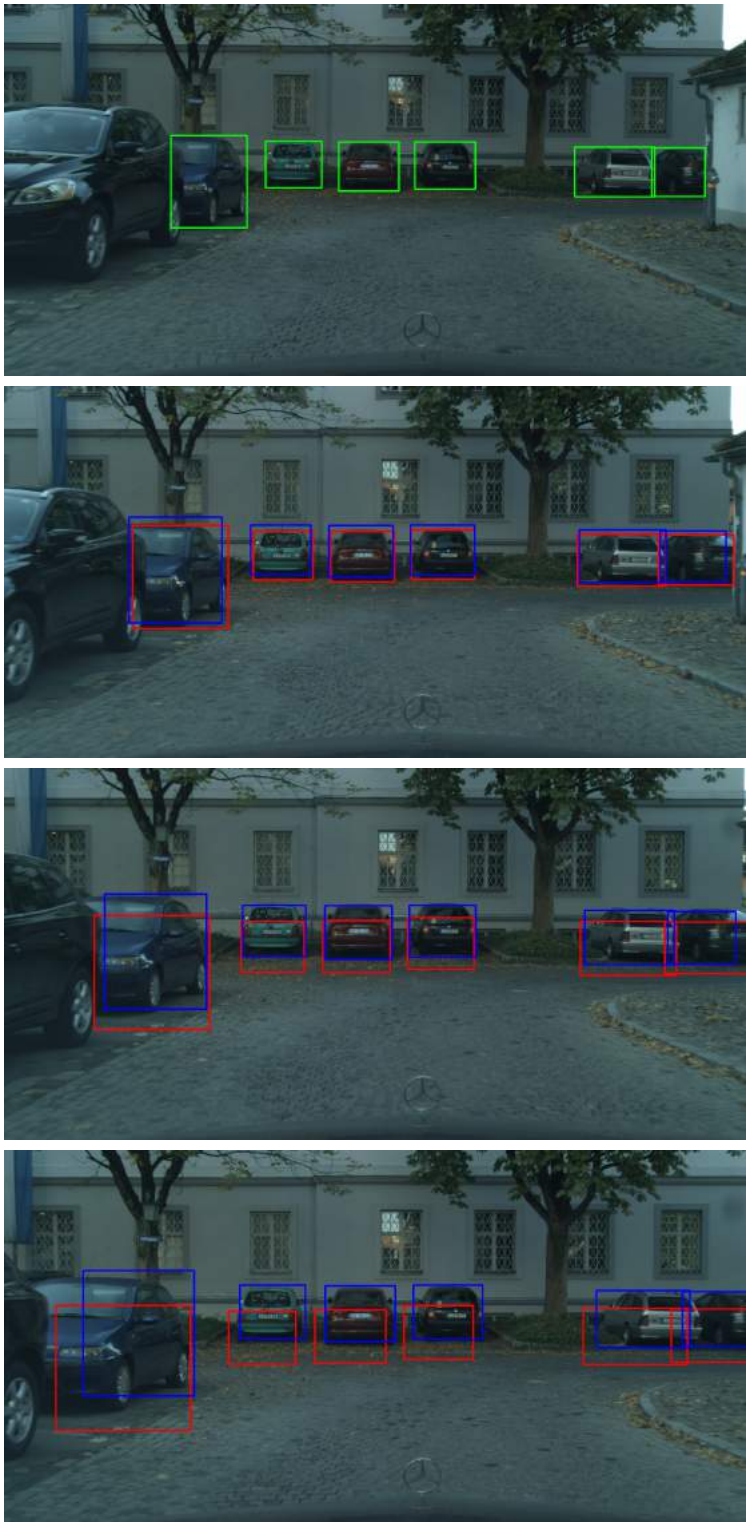


Figure 5.3: Benefit of re-projection for bounding box forecasting. In green the observed bounding boxes at time $t = 0$, in red and blue the prediction with and without re-projection, respectively, at time $t = 6, 12, 18$.

5.4.2 Keypoints forecasting

Keypoints are represented as *xy*-coordinates in the scene: $\mathbf{kp} = [x, y]$. We condition these keypoints to their corresponding bounding box, by removing box center and dividing by box size:

$$\mathbf{kp}^*{}_{x,y} = \frac{\mathbf{kp}_{x,y} - (\mathbf{bbox}_{x,y} + \frac{\mathbf{bbox}_{w,h}}{2})}{\mathbf{bbox}_{w,h}}, \tag{5.6}$$

where $\mathbf{bbox}_{x,y}$ is the top-left corner of the bounding box, $\mathbf{bbox}_{w,h}$ its width and height, $\mathbf{kp}_{x,y}$ the position of the keypoint and $\mathbf{kp}^*{}_{x,y}$ the position of the keypoint, conditioned by the bounding box. We use a linear LSTM to estimate the coordinates of each keypoint or position using a sequence-to-sequence model Venugopalan et al. (2015). For each instance, we incorporate the category information as a onehot encoder. Unlike Villegas et al. (2018), we choose to predict these representations in the coordinate space instead of the heatmap spatial space since it is a simpler representation, thus easier to learn by the network. The LSTM network is trained using an ℓ_2 loss between the prediction and the targets. At test time, the keypoints locations are reconstructed as follow:

$$\hat{\mathbf{kp}}_{x,y} = \hat{\mathbf{kp}}^*{}_{x,y}\hat{\mathbf{bbox}}_{w,h} + \hat{\mathbf{bbox}}_{x,y} + \frac{\hat{\mathbf{bbox}}_{w,h}}{2} \tag{5.7}$$

where $\hat{\mathbf{kp}}^*$ is the predicted keypoint conditioned to object bounding box, and $\hat{\mathbf{bbox}}$ is the predicted object bounding box.

5.4.3 Instance mask forecasting: occlusion aware neural renderer

To predict future instance segmentation we use a similar model as Riochet et al. (2020b), predicting each instance individually and applying an occlusion predictor generating the appropriate pattern of inter-object occlusions. Compared to Riochet et al. (2020b), we enrich the input object representation with its last observed mask, allowing to predict much more complex object shapes.

The Object rendering network takes as input the bounding box $\mathbf{bbox}_t^k$, category $\mathbf{l}^k$ and previous binary mask M_{t-1}^k of object k at time t. The previous instance mask M_{t-1}^k is centered in the bounding box $\mathbf{bbox}_t^k$ and concatenated to a features map encoding for object category: a $C \times H(=128) \times W(=256)$ binary map where each pixel is the $\mathbf{l}$ vector of length C. If object depth is available, it is copied into another $H \times W$ array and concatenated to the feature map. Similarly, if object keypoints are available, they are represented as a binary $H \times W$ array filled with 1 at the location of every keypoint, and concatenated to the feature map.

The input feature map is processed with eight 3×3 convolution filters with a pyramidal number of channels (going from 16 to 256 and backward), padding of size 1 and interlaced with `ReLU` activation functions. The last convolution outputs a $2 \times H \times W$ array, the first channel being the predicted instance mask $\hat{M}^k$, the second encoding its depth $\hat{D}^k$. Note that if ground-truth object depth is not available, the relative position of differents objects to render can be inferred by the model given for example its size, category, keypoints, etc.

The object rendering network is applied to all objects present, resulting in a set of masks and depth representation, denoted as $\{(\hat{M}^k, \hat{D}^k), k = 1..N\}$.

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

$$w_{i,j}^k = \frac{e^{-\lambda \hat{D}_{i,j}^k}}{\sum_{q=1}^{N} e^{-\lambda \hat{D}_{i,j}^q}}, k = 1..N, \tag{5.8}$$

where λ is a parameter learned by the model. The final masks are computed as a weighted combination of masks $\hat{M}_{i,j}^k$ for individual objects k: $\hat{M}_{i,j}^k = w_{i,j}^k \hat{M}_{i,j}^k$, where i, j are output pixel coordinates and $w_{i,j}^k$ the weights given by Eq. 5.8. The intuition is that the occlusion renderer constructs the final output $\hat{M}$ by selecting, for every pixel, the mask with minimal depth (corresponding to the object occluding all other objects), and discarding all other objects at this pixel.

5.4.4 Training details

We train all models with Adam optimizer and its default parameters. For bounding boxes and keypoints predictions, we use a batch size of 3 sequences, which includes a variable number of detected objects, for 400 epochs. For instance mask prediction, we use batch size of 2 and train for 100 epochs. Following Luc et al. (2017), we downsample the segmentation representations to 256×128. On Carla our models are trained to predict 12 outputs from 8 inputs, and on Cityscapes, 5 outputs from 4 inputs. We augment data using horizontal flips with probability 0.5.

5.5 Results

We assess our performance by measuring the mean Intersection over Union (IoU) and mean Average Precision averaged over IOU (AP-50) for segmentations, and the Euclidean distance for keypoints predictions, against ground truth annotations. We begin by validating our approach on the Carla dataset using synthetic annotations, and then show that our results hold on Cityscapes without relying to ground truths.

5.5.1 Ablation studies on Carla

We first highlight the importance of every component of our approach in Table 5.2. Specifically, we remove successively for instance segmentation forecasting: (i) the re-projection step, (ii) the bounding box input, (iii) the keypoint input and report the performance in term of IoU. We first note a large performance gap with the copy baseline that consists in predicting the last observed input, showing the difficulty of the long range predictions we aim to perform. Conditioning on keypoints helps, improving the IoU by two points, and our geometric re-projection helps considerably, bringing almost 6 points. Finally, we also demonstrate the advantage offered by our interaction network modeling by comparing to a ConvLSTM baseline that we detail in the Sup. Mat.

The advantage of conditioning keypoint predictions using bounding boxes is quantified in Table 5.3. Here, we observe that providing boxes in inputs to forecast keypoints helps reduce the error by 57%. The error that is reached is close to the

	IoU
Copy baseline	23.5
Conv-LSTM baseline	44.1
Full model	**54.3**
without keypoints	52.3
without projection	48.4

Table 5.2: Instance forecasting ablation study on Carla, in terms of IoU of moving objects, of the 12^{th} predicted masks. The full model uses geometric re-projection, the Recurrent Interaction Network and keypoints to perform instance masks predictions.

	pix. err.
Standard LSTM	64
GT bbox conditioned LSTM	22
Predicted bbox conditioned LSTM	27

Table 5.3: Keypoints forecasting error using conditioning: average Euclidean distance between prediction and Ground Truth, in pixels, in the 960×540 frame.

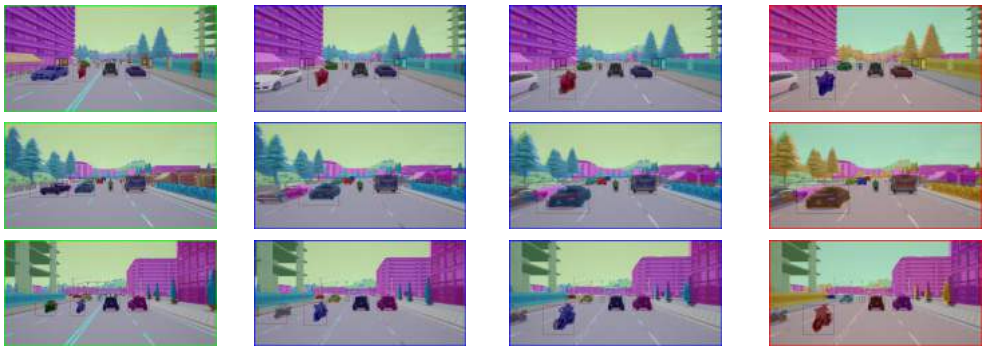
optimal we could get by using Ground truth bounding boxes.

Qualitative examples on Carla are provided in Figure 5.4, that presents predictions with and without using our perspective correction and conditioning. We display, for 3 sequences, the last input at time t, predictions at times $t + 6$ and $t + 12$ without using keypoint conditioning nor re-projection, and the result at $t + 12$ using the full pipeline. We observe that using keypoints and re-projection helps achieve more accurate prediction, in particular on the forehead object of each sequence.

5.5.2 Results on Cityscapes

We adopt a similar setting to Luc et al. (2017) and following works that use inputs from frames 8, 11, 14, 17 for short term prediction of frame 20, and 2, 5, 8, 11 for mid term prediction of frames 14, 17 and 20. In these experiments, we do not use keypoint conditioning and leave this study as future work.

In Table 5.4 we compare the IoU on moving objects, compared with previous



Last input t | Pred w/o kp+proj $t+6$ | Pred w/o kp+proj $t+12$ | Pred with kp+proj $t+12$

Figure 5.4: Comparison of results on Carla with or without help from keypoints conditioning and re-projection. Results are overlayed with ground truth images to emphasise different predictions.

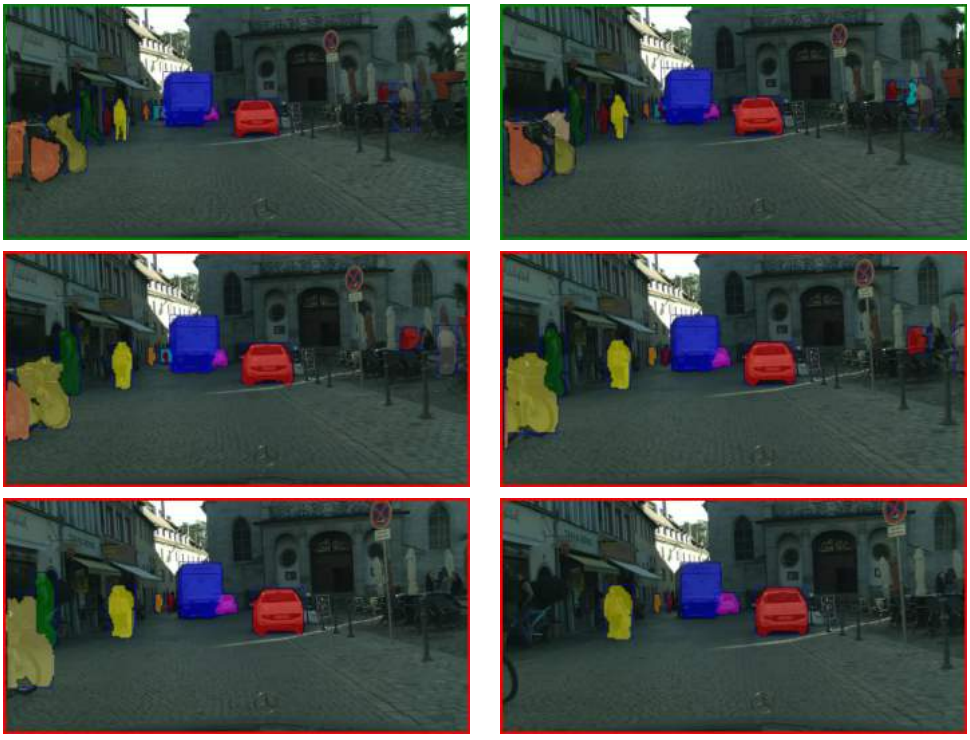


Figure 5.5: Output example on a video sequence from Cityscapes. The model takes as input panoptic segmentations for frame at $t = 2, 5, 8, 11$ (the first two frames beeing omitted here) and predict forward segmentation for frames at $t = 14, 17, 20, 23$.

	IoU Moving Objects	
	Short term	Mid term
Copy last input	48.	29.7
Oracle	64.7	
Luc et al ICCV'17	55.3	40.8
Saric et al. (2019) Oracle	71.5	
Saric et al. (2019)	63.8	49.9
Saric et al. (2020) Oracle	75.2	
Saric et al. (2020)	67.7	**54.6**
D2 oracle	75.7	
Ours	**69.5**	49.5
Ours, no re-proj	67.3	44.7

Table 5.4: Instance forecasting: Mean IoU MO (Moving objects) on cityscapes val set. Comparison to semantic segmentation forecasting approaches.

approaches. We note that the other methods of this table focus on semantic segmentation forecasting and do not delineate instance contours nor track object identities. In terms of moving objects, our method improves the state-of-the-art Saric et al. (2020) for the short-term semantic segmentation prediction.

5.6 Conclusion

We introduce a novel multi-representational forecasting pipeline that builds upon different visual semantic inputs. The modeling of object trajectories via interaction networks helps achieve results that are outperforming the state of the art for the challenging task of future instance segmentation. We hope by providing such a generic approach and analyzing a novel aspect of the forecasting problem to serve as a building block for a better state modeling, and hence realistic forecasting. We invite the reader to view examples of our predictions in the supplementary materials.

5.7 Appendix

5.7.1 Data collection

We adapted the Carla research driving simulator of Dosovitskiy et al. (2017) to generate sequences of images together with their corresponding semantic labels in the form of background segmentations, instance trajectories and masks, and instance keypoints.

The simulator includes five open-world environments approximating real-life driving conditions and allows for randomization of traffic and weather conditions. We show two examples of maps used in our simulations in Figure 5.6.

Map 1

Map 2

Figure 5.6: Examples of environment maps from the Carla driving simulator.

Vehicles in Carla belong to one of 42 different models that can be assigned random colors increasing the variability of the scene. Similarly, a pedestrian belongs to one of five types, of which each vary in measurement characteristics. At each run simulation, we spawn 300 vehicles, pedestrians and bicycles at a randomly selected map. Each of the agents (vehicles, bicycles, pedestrians) is moving autonomously, with 50 of the vehicles having installed cameras that record their surroundings in

an egocentric manner. We show an example of the trajectories created by spawning bicycles, vehicles and pedestrians on Map 1 in Figure 5.7.

Figure 5.7: Simulated trajectories of pedestrians, cyclists, and vehicles in the first map.

For each of the maps, we ran simulations of length 120 time-steps, that is recorded by 50 cameras simultaneously and provides GT labels for many attributes including semantic segmentation, instance segmentation, tracking information, keypoint detection, position and bounding boxes. The camera vehicles are chosen randomly among the vehicles that are spawned in the simulation.

We extracted a variable number of keypoints per agent category: 5 for vehicles, 15 for bicycles, and 12 for pedestrians. For pedestrians, we used MaskRCNN and selected the subset of keypoints that is shared with the ones defined in Carla. Thus, results on Carla represent all moving agents category, while results on Detectron labels only considers the keypoints of humans.

To obtain automatic labels, we ran the Detectron2 predictors for panoptic segmentation and human keypoints on both Cityscapes dataset and our generated Carla data (see Figure 5.8 for the panoptic segmentation predictions). Automatic labels provide weak supervision for the same attributes provided in the GT labels.

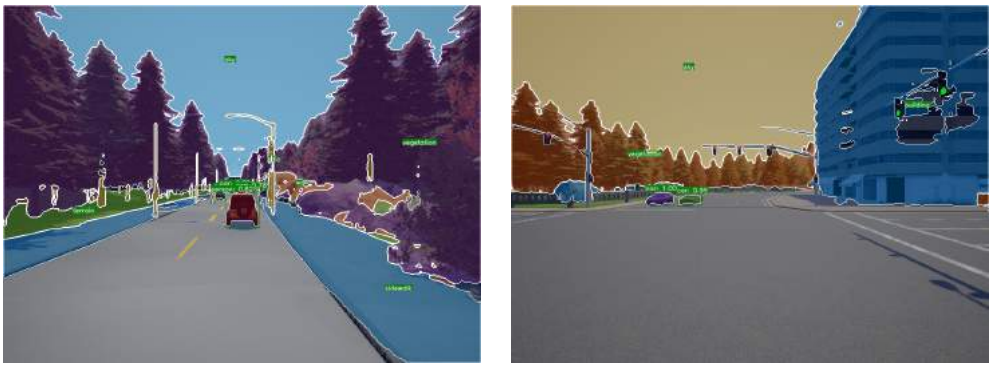


Figure 5.8: Examples of panoptic segmentation results using Detectron2 on Carla frames.

5.7.2 Correcting forward predictions from ego-motion

We focus on predicting future location of objects in video frames, in the case where the camera is moving. We want to take account for camera displacement in order to improve predictions.

Setup

The camera parameters are defined by the camera's intrinsic properties:

- H frame height in pixels (e.g. 540 on Carla)
- W frame width in pixels (e.g. 960 on Carla)
- F camera fovea angle ($\pi/4$)

and its position/orientation:

- $\mathbf{c} = [c_x, c_y, c_z]$ absolute location of camera in the world
- $\theta = [\theta_x, \theta_y, \theta_z]$ camera orientation as [roll, pitch, yaw].

Object are defined by

- $[x, y, z]$ absolute location
- $[p_x, p_y]$ relative location in number of pixels, $[0, 0]$ being the top-left corner
- d depth or distance to the camera

Perspective projection

In this section we describe the perspective projection, which computes (p_x, p_y, d) from (x, y, z), $\mathbf{c}$, θ:

$$(p_x, p_y, d) = \texttt{pproj}_{\mathbf{c},\theta}(x, y, z)$$

- Step 0 (done one time only): given camera intrinsic properties, compute calibration matrix:

$$\mathbf{K} = \begin{bmatrix} \frac{W}{2\tan(F)} & 0 & 0 \\ 0 & \frac{W}{2\tan(F)} & 0 \\ \frac{W}{2} & \frac{H}{2} & 0 \end{bmatrix} \tag{5.9}$$

This linear application "rescales" relative object location: $[-1, 1]^2 \to [0, W] \times [0, H]$.

– Step 1: given camera rotation θ, compute the camera transform:

$$\mathbf{M}(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & \sin(\theta_x) \\ 0 & -\sin(\theta_x) & \cos(\theta_x) \end{bmatrix} \begin{bmatrix} \cos(\theta_y) & 0 & -\sin(\theta_y) \\ 0 & 1 & 0 \\ \sin(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix} \begin{bmatrix} \cos(\theta_z) & \sin(\theta_z) & 0 \\ -\sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.10}$$

This linear application consists in three consecutive rotations, each one along one axis.

– Step 2: given camera location (c_x, c_y, c_z) and object location (x, y, z), compute the intermediate quantity:

$$\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} = \mathbf{K} \times \mathbf{M}(\theta) \times \left(\begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} \right) \tag{5.11}$$

– Step 3: the result is given by:

$$\begin{bmatrix} p_x \\ p_y \\ d \end{bmatrix} = \begin{bmatrix} f_x / f_z \\ f_y / f_z \\ f_z \end{bmatrix} \tag{5.12}$$

Inverse perspective projection

Matrices $\mathbf{K}$ and $\mathbf{M}(\theta)$ are invertible, so is $\texttt{pproj}_{\mathbf{c},\theta}()$. We write the inverse perspective projection:

$$(x, y, z) = \texttt{pproj}^{-1}_{\mathbf{c},\theta}(p_x, p_y, d).$$

5.7.3 Additional ablation for segmentation prediction

To compare segmentation results obtained with our renderer with a classical convLSTM, we implement the alternative architecture described in Figure 5.9.

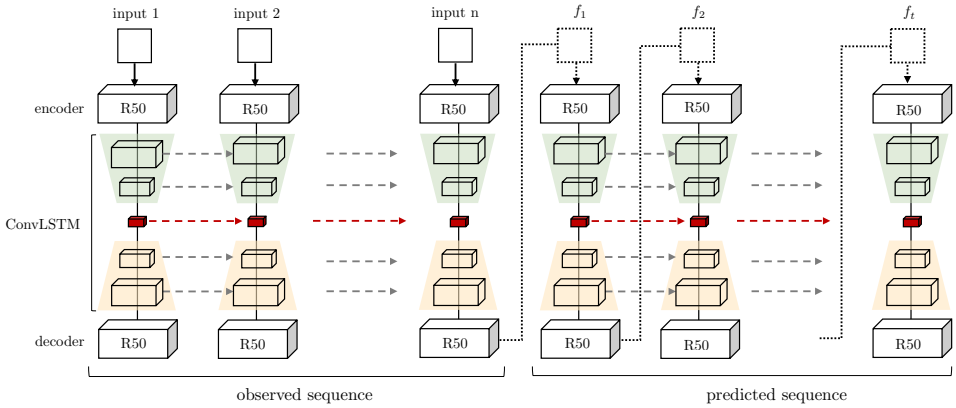


Figure 5.9: Architecture of the spatial prediction module used in our ablation studies. A convolutional R50 encoder followed by a recurrent ConvLSTM that propagates spatio-temporal features that are finally converted back to spatial domain using a deconvolutional R50 decoder.

It represents the spatial encoder-decoder Conv-LSTM architecture we use in our ablations for the background and instance prediction modules. The encoder architecture is a non-recurrent ResNet-50 applied to each frame independently. The extracted spatial features are jointly processed with the spatio-temporal features that are being propagated through time by the Conv-LSTM. Then, the output spatio-temporal features for each step are processed by a decoder (deconvolutional ResNet-50). We used a convolutional LSTM with the following number of hidden channels: 16, 32, 64, 32, 16.

Chapter 6

Conclusion

In this thesis we attempted to draw a bridge between Infant Development and Artificial Intelligence studies on intuitive physics. After introducing challenges and states of the art in both litteratures, we described in chapter 3 a consistent series of tests to evaluate intuitive physics in artificial intelligence systems, the same way cognitive scientists evaluate it among infants.

This benchmark, called IntPhys (see chapter 3), is based on the Violation Expectation Paradigm and investigates three intuitive physics concepts: *object permanence*, *shape consistency* and *trajectory continuity*. We also conducted human performances and compared with two pixel-based baseline models. In the last 3 years, we counted more than 20 teams which evaluated their models on the benchmark, and two other teams used a similar Violation Expectation Paradigm to evaluate Intuitive Physics in systems (Piloto et al. (2018); Smith et al. (2019)).

Our experiments on IntPhys showed that pixel-based CNN encoder-decoder structures - with no accountability for object instances - struggle to learn the type of physical regularities we consider. This is especially true for predicting long trajectories with frequent occlusions. In chapter 4, we designed an object-based model, gifting the system with a notion of *objects*, interacting together and which physics is compositional. Our experiments on simulated videos showed we were able to perform object tracking and forward modelling, even when there were frequent occlusions. In chapter 5, we adapted this approach to predict future instance masks in city driving videos. We showed that decoupling objects' position and appearance allows to predict longer sequences. In addition, we proposed a method to decouple ego-motion from objects' motion, making it easier to learn long term object

dynamics.

To disentangle the appearance of objects and their motion, we relied on object detectors that are pre-trained on hundreds of thousands images with extensive annotation. Infants, of course, don't need such data to acquire intuitive physics and it seems likely that the mechanisms described in introduction still work with categories of objects they have not been used to before. How could we adapt these methods to a broader class of objects? And what is exactly an "object"? We used the notion of "object" that is the commonly considered in computer vision, but a body itself can be seen with various levels of granularities. Our understanding of physics looks to be more hierarchical, and to adapt to more complex scenarios, including soft bodies, liquids, etc. Finally, other senses seem to play an important role in our perception of the physical world (e.g., touch, hearing, proprioception); these have not been explored in this thesis, but I hope it will encourage future students to spend their own on it.

Bibliography

A. Agarwal, S. Gupta, and D. K. Singh. Review of optical flow technique for moving object detection. In *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, pages 409–413, December 2016. doi: 10.1109/IC3I.2016.7917999.

Pulkit Agrawal, Ashvin Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to Poke by Poking: Experiential Learning of Intuitive Physics. *CoRR*, abs/1606.07419, 2016.

Andréa Aguiar and Renée Baillargeon. 2.5-Month-Old Infants' Reasoning about When Objects Should and Should Not Be Occluded. *Cognitive Psychology*, 39(2): 116–157, September 1999. ISSN 00100285. doi: 10.1006/cogp.1999.0717.

Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In *CVPR*, 2016.

Kelsey R. Allen, Kevin A. Smith, and Joshua B. Tenenbaum. Rapid trial-and-error learning with simulation supports flexible tool use and physical reasoning. *arXiv:1907.09620 [cs]*, June 2020.

Oron Ashual and Lior Wolf. Specifying Object Attributes and Relations in Interactive Scene Generation. In *ICCV*, 2019.

R. Baillargeon and S. Carey. Core Cognition and Beyond. In S. Pauen, editor, *Early Childhood Development and Later Outcome*, pages 33–65. Cambridge University Press, New York, 2012.

Renée Baillargeon. Object permanence in $3\frac{1}{2}$- and $4\frac{1}{2}$-month-old infants. *Developmental Psychology*, 23:655–664, September 1987. doi: 10.1037//0012-1649.23.5.655.

Renee Baillargeon and Stephanie Hanko-Summers. Is the top object adequately supported by the bottom object? Young infants' understanding of support relations. *Cognitive Development*, 5(1):29–53, 1990.

Renée Baillargeon, Elizabeth Spelke, and Stan Wasserman. Object Permanence in Five-Month-Old Infants. *Cognition*, 20:191–208, September 1985. doi: 10.1016/0010-0277(85)90008-3.

Renee Baillargeon, Amy Needham, and Julie DeVos. The development of young infants' intuitions about support. *Infant and Child Development*, 1(2):69–78, 1992.

Anton Bakhtin, Laurens van der Maaten, Justin Johnson, Laura Gustafson, and Ross Girshick. PHYRE: A New Benchmark for Physical Reasoning. *arXiv:1908.05656 [cs, stat]*, August 2019.

Fabien Baradel, Natalia Neverova, Julien Mille, Greg Mori, and Christian Wolf. COPHY: Counterfactual Learning of Physical Dynamics. *arXiv:1909.12000*, 2019.

Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems*, pages 4502–4510, 2016.

Peter W. Battaglia, Jessica B. Hamrick, and Joshua B. Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences of the United States of America*, 110(45), 2013.

Luca Bertinetto, Jack Valmadre, João F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision*, pages 850–865. Springer, 2016.

Margrit Betke, Esin Haritaoglu, and Larry S. Davis. Real-time multiple vehicle detection and tracking from a moving vehicle. *Machine Vision and Applications*, 12(2):69–83, August 2000. ISSN 1432-1769. doi: 10.1007/s001380050126.

Apratim Bhattacharyya, Mario Fritz, and Bernt Schiele. Long-Term On-Board Prediction of People in Traffic Scenes Under Uncertainty. In *CVPR*, 2018.

Christopher P. Burgess, Loïc Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matthew M. Botvinick, and Alexander Lerchner. MONet: Unsupervised Scene Decomposition and Representation. *ArXiv*, 2019.

Susan Carey. *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.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Michael B Chang, Tomer Ullman, Antonio Torralba, and Joshua B Tenenbaum. A compositional object-based approach to learning physical dynamics. *arXiv preprint arXiv:1612.00341*, 2016.

Erik W. Cheries, Stephen R. Mitroff, Karen Wynn, and Brian J. Scholl. Cohesion as a constraint on object persistence in infancy. *Developmental Science*, 11(3): 427–432, 2008. ISSN 1467-7687. doi: 10.1111/j.1467-7687.2008.00687.x.

Hsu-Kuang Chiu, Ehsan Adeli, Borui Wang, De-An Huang, and Juan Carlos Niebles. Action-Agnostic Human Pose Forecasting. *WACV*, 2019.

Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. *arXiv preprint arXiv:1604.00449*, 2016.

Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial Video Generation on Complex Datasets. In *arXiv:1907.06571v2*, 2019.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *CVPR*, 2016.

Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. 2016.

Camille Couprie, Pauline Luc, and Jakob Verbeek. Joint Future Semantic and Instance Segmentation Prediction. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

Emily L. Denton, Sam Gross, and Rob Fergus. Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks. *CoRR*, abs/1611.06430, 2016.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. *Conference on Robot Learning*, 2017.

Emmanuel Dupoux. Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner. *Cognition*, 173:43–59, April 2018. ISSN 0010-0277. doi: 10.1016/j.cognition.2017.11.008.

Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual Foresight: Model-Based Deep Reinforcement Learning for Vision-Based Robotic Control. *arXiv:1812.00568 [cs]*, December 2018.

Sébastien Ehrhardt, Aron Monszpart, Niloy J. Mitra, and Andrea Vedaldi. Learning A Physical Long-term Predictor. *CoRR*, abs/1703.00247, 2017a.

Sebastien Ehrhardt, Aron Monszpart, Niloy J. Mitra, and Andrea Vedaldi. Taking Visual Motion Prediction To New Heightfields. 2017b.

Epic Games. Unreal engine, April 2019.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

Ali Farhadi, Mohsen Hejrati, Mohammad Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. *Computer vision–ECCV 2010*, pages 15–29, 2010.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *Robotics and Automation (ICRA), 2017 IEEE International Conference On*, pages 2786–2793. IEEE, 2017.

Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems*, pages 64–72, 2016.

R. Fox, R. N. Aslin, S. L. Shea, and S. T. Dumais. Stereopsis in human infants. *Science*, 207(4428):323–324, January 1980.

Marco Fraccaro, Simon Kamronn, Ulrich Paquet, and Ole Winther. A Disentangled Recognition and Nonlinear Dynamics Model for Unsupervised Learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3601–3610. Curran Associates, Inc., 2017.

Katerina Fragkiadaki, Pulkit Agrawal, Sergey Levine, and Jitendra Malik. Learning visual predictive models of physics for playing billiards. *arXiv preprint arXiv:1511.07404*, 2015a.

Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *ICCV*, 2015b.

Fadri Furrer, Martin Wermelinger, Hironori Yoshida, Fabio Gramazio, Matthias Kohler, Roland Siegwart, and Marco Hutter. Autonomous robotic stone stacking with online next best object target pose planning. pages 2350–2356, May 2017. doi: 10.1109/ICRA.2017.7989272.

Charles R. Gallistel. *The Organization of Learning*. The Organization of Learning. The MIT Press, Cambridge, MA, US, 1990. ISBN 978-0-262-07113-0.

R. Girshick. Fast R-CNN. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, December 2015. doi: 10.1109/ICCV.2015.169.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *2014*

IEEE Conference on Computer Vision and Pattern Recognition, pages 580–587, Columbus, OH, USA, June 2014. IEEE.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Neural Expectation Maximization. *arXiv:1708.03498 [cs, stat]*, November 2017.

Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Chris Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-Object Representation Learning with Iterative Variational Inference. *arXiv:1903.00450 [cs, stat]*, March 2019.

Oliver Groth, Fabian B. Fuchs, Ingmar Posner, and Andrea Vedaldi. ShapeStacks: Learning Vision-Based Physical Intuition for Generalised Object Stacking. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, volume 11205, pages 724–739. Springer International Publishing, Cham, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *CVPR*, 2017.

Mikael Henaff, Alfredo Canziani, and Yann LeCun. Model-Predictive Policy Learning with Uncertainty Regularization for Driving in Dense Traffic. In *ICLR*, 2019.

Susan Hespos and Kristy Vanmarle. Physics for infants: Characterizing the origins of knowledge about objects, substances, and number. *Wiley Interdisciplinary Reviews-Cognitive Science*, 3:19–27, January 2012. doi: 10.1002/Wcs.157.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997. ISSN 0899-7667.

Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Li F Fei-Fei, and Juan Carlos Niebles. Learning to Decompose and Disentangle Representations for Video Prediction. page 10.

Ashesh Jain, Amir Roshan Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-RNN: Deep Learning on Spatio-Temporal Graphs. *CVPR*, 2016.

William James. *The Principles of Psychology*. Holt, 1890.

Michael Janner, Sergey Levine, William T. Freeman, Joshua B. Tenenbaum, Chelsea Finn, and Jiajun Wu. Reasoning About Physical Interactions with Object-Oriented Prediction and Planning. In *ICLR*, 2019.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. *arXiv preprint arXiv:1612.06890*, 2016.

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

In Kyeong Kim and Elizabeth S. Spelke. Infants' sensitivity to effects of gravity on visible object motion. *Journal of Experimental Psychology: Human Perception and Performance*, 18(2):385–393, 1992. ISSN 1939-1277(Electronic),0096-1523(Print). doi: 10.1037/0096-1523.18.2.385.

Yunji Kim, Seonghyeon Nam, In Cho, and Seon Joo Kim. Unsupervised Keypoint Learning for Guiding Class-Conditional Video Prediction. *NeurIPS*, 2019.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980, 2014.

Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic Feature Pyramid Networks. *arXiv preprint arXiv:1901.02446*, 2019.

Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *ECCV*, 2012.

Dieter Koller, Joseph Weber, and Jitendra Malik. Robust multiple car tracking with occlusion reasoning. In Jan-Olof Eklundh, editor, *Computer Vision — ECCV '94*, Lecture Notes in Computer Science, pages 189–196, Berlin, Heidelberg, 1994. Springer. ISBN 978-3-540-48398-4. doi: 10.1007/3-540-57956-7_22.

Hema S Koppula and Ashutosh Saxena. Anticipating human activities using object affordances for reactive robotic response. *Trans. on Pattern analysis and machine intelligence*, 38(1):14–29, 2015.

Adam Kosiorek, Hyunjik Kim, Yee Whye Teh, and Ingmar Posner. Sequential Attend, Infer, Repeat: Generative Modelling of Moving Objects. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 8606–8616. Curran Associates, Inc., 2018.

Matej Kristan, Jiri Matas, Aleš Leonardis, Tomas Vojir, Roman Pflugfelder, Gustavo Fernandez, Georg Nebehay, Fatih Porikli, and Luka Čehovin. *A Novel Performance Evaluation Methodology for Single-Target Trackers*. January 2016.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

Tian Lan, Tsung-Chuan Chen, and Silvio Savarese. A Hierarchical Representation for Future Action Prediction. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, Lecture Notes in Computer Science, pages 689–704, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10578-9. doi: 10.1007/978-3-319-10578-9_45.

Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 667–676, 2017.

Adam Lerer, Sam Gross, and Rob Fergus. Learning Physical Intuition of Block

Towers by Example. In *International Conference on Machine Learning*, pages 430–438, June 2016.

Adam Kal Lerer, Robert D. Fergus, and Ronan Alexandre Riochet. *Differentiating Physical and Non-Physical Events*. Google Patents, October 2019.

Guanbin Li, Yuan Xie, Tianhao Wei, Keze Wang, and Liang Lin. Flow Guided Recurrent Neural Encoder for Video Salient Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3243–3252, 2018a.

Wenbin Li, Ales Leonardis, and Mario Fritz. To Fall Or Not To Fall: A Visual Approach to Physical Stability Prediction. *arXiv preprint*, 2016.

Wenbin Li, Ales Leonardis, and Mario Fritz. Visual stability prediction for robotic manipulation. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2606–2613, Singapore, Singapore, May 2017. IEEE.

Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B. Tenenbaum, and Antonio Torralba. Learning Particle Dynamics for Manipulating Rigid Bodies, Deformable Objects, and Fluids. In *International Conference on Learning Representations*, September 2018b.

Xiaodan Liang, Lisa Lee, Wei Dai, and Eric P. Xing. Dual Motion GAN for Future-Flow Embedded Video Prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1744–1752, 2017.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, Lecture Notes in Computer Science, pages 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1. doi: 10.1007/978-3-319-10602-1_48.

Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *Proceedings*

of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2117–2125, 2017.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535, December 2016. doi: 10.1162/tacl_a_00115.

Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the CoordConv solution. page 12.

Y. Liu, H. Zheng, X. Feng, and Z. Chen. Short-term traffic flow prediction with Conv-LSTM. In *2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–6, October 2017. doi: 10.1109/WCSP.2017.8171119.

John Locke. *An Essay Concerning Human Understanding*. 1689.

William Lotter, Gabriel Kreiman, and David Cox. Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning. *arXiv:1605.08104 [cs, q-bio]*, February 2017.

Wei-Lwun Lu, Jo-Anne Ting, James J. Little, and Kevin P. Murphy. Learning to Track and Identify Players from Broadcast Sports Videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1704–1716, July 2013. ISSN 1939-3539. doi: 10.1109/TPAMI.2012.242.

Pauline Luc, Natalia Neverova, Camille Couprie, Jakob Verbeek, and Yann LeCun. Predicting Deeper Into the Future of Semantic Segmentation. In *ICCV*, 2017.

Pauline Luc, Camille Couprie, Yann LeCun, and Jakob Verbeek. Predicting Future Instance Segmentation by Forecasting Convolutional Features. In *ECCV*, 2018.

Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, Xiaowei Zhao, and Tae-Kyun Kim. Multiple Object Tracking: A Literature Review. *arXiv:1409.7618 [cs]*, May 2017.

Yuexin Ma, Xinge Zhu, Sibo Zhang, Ruigang Yang, Wenping Wang, and Dinesh Manocha. TrafficPredict: Trajectory Prediction for Heterogeneous Traffic-Agents. *AAAI*, 2019.

Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

Rowan McAllister, Yarin Gal, Alex Kendall, Mark Van Der Wilk, Amar Shah, Roberto Cipolla, and Adrian Vivian Weller. Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. International Joint Conferences on Artificial Intelligence, Inc., 2017.

Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. *CoRR*, abs/1411.1784, 2014.

Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Generalizable features from unsupervised learning. *ICLR Workshop submission*, 2017.

Roozbeh Mottaghi, Hessam Bagherinezhad, Mohammad Rastegari, and Ali Farhadi. Newtonian Image Understanding: Unfolding the Dynamics of Objects in Static Images. *Computer Vision and Pattern Recognition (CVPR)*, 2016.

Damian Mrowca, Chengxu Zhuang, Elias Wang, Nick Haber, Li F Fei-Fei, Josh Tenenbaum, and Daniel L Yamins. Flexible neural representation for physics prediction. In *Advances in Neural Information Processing Systems*, pages 8799–8810, 2018.

Seyed Nabavi, Mrigank Rochan, and Yang Wang. Future Semantic Segmentation with Convolutional LSTM. *BMVC*, 2018.

Amy Needham. The role of shape in 4-month-old infants' object segregation. *Infant Behavior and Development*, 22(2):161–178, January 1999. ISSN 01636383.

Eric Neufeld. Review of Statistical methods for speech recognition by Frederick Jelinek. The MIT Press 1997. *Computational Linguistics*, 25:297–298, June 1999.

Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games. In *Advances in Neural Information Processing Systems*, pages 2863–2871, 2015.

Viorica Pătrăucean, Ankur Handa, and Roberto Cipolla. Spatio-temporal video autoencoder with differentiable memory. In *International Conference on Learning Representations (ICLR) Workshop*, 2016.

S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pages 261–268, September 2009. doi: 10.1109/ICCV.2009.5459260.

Jean Piaget. *The Construction of Reality in the Child*. The Construction of Reality in the Child. Basic Books, New York, NY, US, 1954. doi: 10.1037/11168-000.

Luis Piloto, Ari Weinstein, Dhruva TB, Arun Ahuja, Mehdi Mirza, Greg Wayne, David Amos, Chia-Chun Hung, and Matt M. Botvinick. Probing Physics Knowledge Using Tools from Developmental Psychology. *CoRR*, abs/1804.01128, 2018.

Pedro O Pinheiro, Ronan Collobert, and Piotr Dollar. Learning to segment object candidates. In *Advances in Neural Information Processing Systems*, pages 1990–1998, 2015.

Steven Pinker. The bootstrapping problem in language acquisition. In *Mechanisms of Language Aquisition*, pages 399–441. Lawrence Erlbaum Associates, Inc, Hillsdale, NJ, US, 1987. ISBN 978-0-89859-596-3 978-0-89859-973-2.

Zenon W Pylyshyn and Ron W Storm. Tracking multiple independent targets: Evidence for a parallel tracking mechanism. *Spatial vision*, 3(3):179–197, 1988.

Xiaojuan Qi, Zhengzhe Liu, Qifeng Chen, and Jiaya Jia. 3D Motion Decomposition for RGBD Future Dynamic Scene Synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7673–7682, 2019.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *CoRR*, abs/1511.06434, 2015.

Syed Ashiqur Rahman and Donald A. Adjeroh. Deep Learning using Convolutional LSTM estimates Biological Age from Physical Activity. *Scientific Reports*, 9(1): 11425, August 2019.

MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: A baseline for generative models of natural videos. *arXiv:1412.6604 [cs]*, May 2016.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, June 2017.

R. Riochet, M. Ynocente Castro, M. Bernard, A. Lerer, R. Fergus, V. Izard, and E. Dupoux. IntPhys: A Framework and Benchmark for Visual Intuitive Physics Reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Ronan Riochet, Mohamed Elfeki, Natalia Neverova, Emmanuel Dupoux, and Camille Couprie. Multi-Representational Future Forecasting. November 2020a.

Ronan Riochet, Josef Sivic, Ivan Laptev, and Emmanuel Dupoux. Occlusion resistant learning of intuitive physics from videos. 2020b.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. ISSN 1573-1405. doi: 10.1007/s11263-015-0816-y.

Josip Saric, Marin Orsic, Tonci Antunovic, Sacha Vrazic, and Sinisa Segvic. Single Level Feature-to-Feature Forecasting with Deformable Convolutions. *CoRR*, abs/1907.11475, 2019.

Josip Saric, Marin Orsic, Tonci Antunovic, Sacha Vrazic, and Sinisa Segvic. Warp to the Future: Joint Forecasting of Features and Feature Motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10648–10657, 2020.

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

Kevin Smith, Lingjie Mei, Shunyu Yao, Jiajun Wu, Elizabeth Spelke, Josh Tenenbaum, and Tomer Ullman. Modeling Expectation Violation in Intuitive Physics with Coarse Probabilistic Object Representations. In *Advances in Neural Information Processing Systems*, pages 8983–8993, 2019.

Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid Dilated Deeper ConvLSTM for Video Salient Object Detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 715–731, 2018.

C. Spampinato, Yun-Heh Chen-Burger, Gayathri Nadarajan, and Robert B. Fisher. Detecting, Tracking and Counting Fish in Low Quality Unconstrained Underwater Videos. In *VISAPP*, 2008. doi: 10.5220/0001077705140519.

E. S. Spelke, K. Breinlinger, J. Macomber, and K. Jacobson. Origins of knowledge. *Psychological Review*, 99(4):605–632, October 1992. ISSN 0033-295X. doi: 10.1037/0033-295x.99.4.605.

Elizabeth S. Spelke, Gary Katz, Susan E. Purcell, Sheryl M. Ehrlich, and Karen Breinlinger. Early knowledge of object motion: Continuity and inertia. *Cognition*, 51(2):131–176, February 1994. ISSN 0010-0277. doi: 10.1016/0010-0277(94)90013-2.

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

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised Learning of Video Representations using LSTMs. In *International Conference on Machine Learning*, pages 843–852. PMLR, June 2015.

Jiangxin Sun, Jiafeng Xie, Jian-Fang Hu, Zihang Lin, Jianhuang Lai, Wenjun Zeng, and Wei-shi Zheng. Predicting future instance segmentation with contextual pyramid convlstms. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2043–2051, 2019.

Mingxing Tan, Ruoming Pang, and Quoc V. Le. EfficientDet: Scalable and Efficient Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10781–10790, 2020.

Unity Technologies. Unity, 2005.

Adam Terwilliger, Garrick Brazil, and Xiaoming Liu. Recurrent flow-guided semantic forecasting. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1703–1712, 2019.

E. Todorov, T. Erez, and Y. Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, October 2012. doi: 10.1109/IROS.2012.6386109.

Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014.

Wei-Chih Tu, Shengfeng He, Qingxiong Yang, and Shao-Yi Chien. Real-Time Salient Object Detection With a Minimum Spanning Tree. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2334–2342, 2016.

Eloisa Valenza, Irene Leo, Lucia Gava, and Francesca Simion. Perceptual Completion in Newborn Human Infants. *Child Development*, 77(6):1810–1821, 2006.

Sjoerd van Steenkiste, Michael Chang, Klaus Greff, and Jürgen Schmidhuber. Relational Neural Expectation Maximization: Unsupervised Discovery of Objects and their Interactions. *arXiv:1802.10353 [cs]*, February 2018.

Rishi Veerapaneni, John D. Co-Reyes, Michael Chang, Michael Janner, Chelsea Finn, Jiajun Wu, Joshua B. Tenenbaum, and Sergey Levine. Entity Abstraction in Visual Model-Based Reinforcement Learning. *arXiv:1910.12827 [cs, stat]*, May 2020.

Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4534–4542, 2015.

Ruben Villegas, Jimei Yang, Yuliang Zou, Sungryull Sohn, Xunyu Lin, and Honglak Lee. Learning to Generate Long-term Future via Hierarchical Prediction. *arXiv:1704.05831 [cs]*, January 2018.

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating Videos with Scene Dynamics. *arXiv:1609.02612 [cs]*, October 2016.

Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking Emerges by Colorizing Videos. *arXiv:1806.09594 [cs]*, July 2018.

Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. The pose knows: Video forecasting by generating pose futures. In *ICCV*, 2017.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-Video Synthesis. *arXiv:1808.06601 [cs]*, December 2018.

Nicholas Watters, Daniel Zoran, Theophane Weber, Peter Battaglia, Razvan Pascanu, and Andrea Tacchetti. Visual Interaction Networks: Learning a Physics Simulator from Video. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4539–4547. Curran Associates, Inc., 2017.

Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling Autoregressive Video Models. In *arXiv:1906.02634*, 2019.

Nevan Wichers, Ruben Villegas, Dumitru Erhan, and Honglak Lee. Hierarchical Long-term Video Prediction without Supervision. *arXiv:1806.04768 [cs]*, June 2018.

Teresa Wilcox. Object individuation: Infants' use of shape, size, pattern, and color. *Cognition*, 72(2):125–166, September 1999. ISSN 0010-0277. doi: 10.1016/S0010-0277(99)00035-9.

John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–227, 2009.

Di Wu, Zhaoyong Zhuang, Canqun Xiang, Wenbin Zou, and Xia Li. 6D-VNet: End-To-End 6-DoF Vehicle Pose Estimation From Monocular RGB Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019a.

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.

Jiajun Wu, Erika Lu, Pushmeet Kohli, Bill Freeman, and Josh Tenenbaum. Learning to see physics via visual de-animation. In *Advances in Neural Information Processing Systems*, pages 152–163, 2017a.

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

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. *Detectron2*. 2019b.

SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, pages 802–810, 2015.

Dejing Xu, Jun Xiao, Zhou Zhao, Jian SHao, Di Xie, and Yueting Zhuang. Self-supervised Spatiotemporal Learning via Video Clip Order Prediction. *CVPR*, 2019a.

Fei Xu and Susan Carey. Infants' metaphysics: The case of numerical identity. *Cognitive psychology*, 30(2):111–153, 1996.

Zhenjia Xu, Jiajun Wu, Andy Zeng, Joshua B. Tenenbaum, and Shuran Song. DensePhysNet: Learning Dense Physical Object Representations via Multi-step Dynamic Interactions. *arXiv:1906.03853 [cs]*, June 2019b.

Tianfan Xue, Jiajun Wu, Katherine Bouman, and Bill Freeman. Visual Dynamics: Probabilistic Future Frame Synthesis via Cross Convolutional Networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 91–99. Curran Associates, Inc., 2016.

Bo Yang, Chang Huang, and Ram Nevatia. Learning affinities and dependencies for multi-target tracking using a CRF model. In *CVPR 2011*, pages 1233–1240, June 2011. doi: 10.1109/CVPR.2011.5995587.

J. Yang and M. Yang. Top-Down Visual Saliency via Joint CRF and Dictionary Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39 (3):576–588, March 2017. ISSN 1939-3539. doi: 10.1109/TPAMI.2016.2547384.

Yu Yao, Mingze Xu, Chiho Choi, David J Crandall, Ella M Atkins, and Behzad Dariush. Egocentric vision-based future vehicle localization for intelligent driving assistance systems. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9711–9717. IEEE, 2019.

Yufei Ye, Maneesh Singh, Abhinav Gupta, and Shubham Tulsiani. Compositional Video Prediction. *ICCV*, 2019.

Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

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. *CogSci*, 2016.

Z. Zhao, P. Zheng, S. Xu, and X. Wu. Object Detection With Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11): 3212–3232, November 2019. ISSN 2162-2388. doi: 10.1109/TNNLS.2018.2876865.

Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-Guided Feature Aggregation for Video Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 408–417, 2017.

C Lawrence Zitnick and Devi Parikh. Bringing semantics into focus using visual abstraction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3009–3016, 2013.

RÉSUMÉ

Pour effectuer des tâches complexes de manière autonome, les systèmes d'intelligence artificielle doivent pouvoir comprendre les interactions physiques entre les objets, afin d'anticiper les conséquences de situations diverses. L'objectif de cette thèse est d'étudier l'acquisition de cette notion (souvent appelée *physique intuitive*), pour un système, de manière autonome et non supervisée à partir de vidéos.
La première contribution consiste en un protocole de test, IntPhys, dont le but est d'évaluer les capacités d'un tel système à comprendre la physique intuitive. Inspiré de la littérature en sciences cognitive, ce protocole consiste à en évaluer la capacité à différencier les évèvement physiques possibles et impossibles au sein de vidéos. Après avoir décrit en détail cette procédure, nous évaluons les performances de deux réseaux de neuronnes convolutifs et les comparons avec les performances humaines.
L'analyse de ces résultats montre les limites des réseaux de neuronnes convolutifs pour prédire la trajectoire des objects à long terme, notamment en présence d'occlusions. Pour cette raison, nous proposons une formulation probabiliste du problème dans laquelle chaque object a sa propre représentation en variables latentes. Nous proposons également une série d'approximations pour trouver une solution acceptable à ce problème d'optimisation.
Dans un dernier chapitre, nous proposons d'appliquer cette approche à un cas pratique: l'anticipation du mouvements des objets alentours lors de la conduite en ville. Nous y montrons qu'il est possible d'entraîner un système à anticiper les futurs masques d'instance d'objets, au sein de séquences vidéos enregistrées lors de la conduite dans 50 villes europénnes.

MOTS CLÉS

Vision par Ordinateur, Physique Intuitive, Apprentissage non-supervisé.

ABSTRACT

To reach human performance on complex tasks, a key ability for artificial intelligence systems is to understand physical interactions between objects, and predict future outcomes of a situation. In this thesis we investigate how a system can learn this ability, often referred to as *intuitive physics*, from videos with minimal annotation.
Our first contribution is an evaluation benchmark, named IntPhys, which diagnoses how much a system understands intuitive physics. Inspired by works in infant development, we propose a Violation-of-Expection procedure in which the system must tell apart well matched videos of possible versus impossible events constructed with a game engine. We describe two Convolutional Neural Networks trained on a forward prediction task, and compare their results with human data acquired with Amazon Mechanical Turk.
The analysis of these results show limitations of CNN encoder-decoders with no structured representation of objects when it comes to predict long-term object trajectories, especially in case of occlusions. In a second work, we propose a probabilistic formulation of learning intuitive physics in 3D scenes with significant inter-object occlusions, in which object positions are modelled as latent variables, enabling the reconstruction of the scene. We propose a series of approximations that make this problem tractable and introduce a compositional neural network demonstrating significant improvements on the intuitive physics benchmark IntPhys. We evaluate this model on a second dataset with increasing levels of occlusions, showing it realistically predicts segmentation masks up to 30 frames in the future.
In a third work, we adapt this approach to a real life application: predicting future instance masks of objects in the Cityscapes Dataset, made of video sequences recorded in streets from 50 cities. We use a state-of-the-art objects detector to estimate object states, then apply the model presented above to predict objects instance masks up to 9 frames in the future. In addition, we propose a method to decouple ego-motion from objects' motion, making it easier to learn long term object dynamics.

KEYWORDS

Computer Vision, Intuitive Physics, Unsupervised Learning.