Learn intuitive physics from videos with multi-scale CGANs

Rong Zhi
TU Darmstadt

Yunlong Song
TU Darmstadt

Boris Belousov
TU Darmstadt

Abstract

Deep neural networks enable transformation of raw pixel inputs to useful feature representations in unsupervised fashion. We investigate the feasibility of using such representations learned from unlabeled videos for predicting trajectories of target objects. Since both the input and the output of the network constitutes a sequence of video frames, the network weights capture the notion of intuitive physics learned from the observed videos. We adapt the multi-scale conditional generative adversarial net (CGAN) architecture, that has been shown to perform well on natural image sequence prediction [1] and used to generate realistically looking videos of Ms. Pac-Man [2], to our settings, and demonstrate that it can successfully predict plausible video sequences that the test object follows a short-range ground truth trajectory.

1 Introduction

Humans learn to understand the physics of the world at a very early age. Evidence suggests that babies develop visual understanding of basic physical concepts during first years of their lives [3]. Endowing a robot with the ability to learn common sense intuitive physics of the real world from visual input is an important step towards the general goal of artificial intelligence.

Learning intuitive physics directly from high-dimensional visual input is a challenging problem that has interested computer vision community for decades. Recently, deep neural networks has been achieving great success on various problems in computer vision, which can be considered as a promising research direction for solving this task. For example, Jiajun Wu, et al. [4] built a physics engine that perceives physical object properties with deep learning method.

In this paper, we address the problem of learning intuitive physics of objects from unlabeled videos by building a deep multi-scale conditional generative adversarial network (CGAN) [1], that is capable of extracting physical features from video sequences and predicting future frames in pixel space. We assume that the better the predictions of the trained generative model, the better the physical representation should be.

Firstly, we build a generative adversarial network that contains a generative model G for next frame generation based on a sequence of input frames and a discriminative model D for estimating the probability that the last frame of the sequence comes from the dataset instead of being produced by the generative model G.

Secondly, we exploit the multi-scale structure of input frames, scaling up the model to a series of structure, each of which captures image structure at a particular scale of an image pyramid.

Finally, we train the G and D simultaneously with part of Physics101 dataset and a tetherball video dataset from our Intelligent Autonomous Systems Labs, and achieve a plausible model that successfully predict a series of frames of high sharpness.

2 Related Work

Deep Learning in Robotics: Deep learning is the fastest-growing field in machine learning and computer vision. Deep models have also been used in robotics. In robotic manipulation, Agrawal, et al. [5] combines deep learning with forward and inverse models to poke objects to desired goal locations. Chelsea Finn, et al. [6] use unsupervised learning strategy for physical interaction through video prediction.

Physics Prediction: Several work have explicitly addressed prediction of physical interactions, including predicting ball motion under forces [9], future human interactions [10], and future car trajectories [11]. Jiajun Wu, et al. [12] proposed a model for learning physical properties (material, mass, density, etc.) from unlabeled videos by encoding basic physical laws into

their model.

Video Prediction: The problem of video prediction is given by a variety of architecture in deep learning. Recently, Mathieu, et al. [1] proposed a deep multi-scale video prediction method by using GANs. Srivastava, et al. [7] applied a Long Short Term Memory (LSTM) networks to learn representations of video sequences, which is based on Autoencoder Model. Our model also constructs a GAN network, but we use it to represent physics.

3 Method

In this section, we will introduce the Generative Adversarial Networks (GANs), which were first introduced by Ian Goodfellow et al. [13] and used for image generation from random noise. The adversarial training approach then was exploited by Michael Mathieu et al. [1] for frame prediction, where a series of future frames were predicted from a sequence of input frames using two networks trained simultaneously. In our project, we use the same approach as proposed by Michael Mathieu et al.

3.1 Generative Adversarial Nets

In GANs [13], a generative model learns to synthesize samples that best resemble the training set, while a discriminative model learns to distinguish between samples drawn from the dataset and samples synthesized by the generator.

We represent the raw video data as a tuple (X, Y):

$$\begin{aligned} X &= \{X^1, X^2, X^3, ..., X^n\} \\ Y &= \{Y^1, Y^2, Y^3, ..., Y^m\} \end{aligned} \tag{1}$$

in which Y^j are pixel matrices of a sequence of M output frames to be predicted from the piexl matrices X^i of N concatenated input frames.

The GANs contain two major models, generative model G and discriminative model D as shown in Figure 1. The generative model G is trained to learn the parameters Θ that best represent the features, based on convolutional neural networks (CNNs) (LeCun et al., 1998), therefore can predict future frames $\hat{Y} = G(X; \Theta_g)$ given the concatenated frames X by minimizing the ℓ_2 norm between the predicted frame and the ground truth frame:

$$\min_G \ell_2(X, Y) = \sqrt{\sum_i (Y_i - \hat{Y}_i)^2} \tag{2}$$

where G represents a differentiable function of the generative models with parameters Θ_g.

The ℓ_2 loss, however, leads to an average result $Y_{\text{avg}} = Y_1 + Y_2$, where Y_1 and Y_2 have the same probability. By introducing a discriminator D, we actually add a standard classifier that can discern between a ground truth frame and a generated frame. More precisely, we train D to classify an input (X, Y) from the dataset into class 1 and generated input $(X, G(Y; \Theta_g))$ to class 0. It turns out that the trained D can be considered as a penalization term to G, so that G can learn the optimal parameters Θ_g and make predictions to confuse D.

We train the D model to maximize the probability of assigning the correct label to both ground truth frames from the dataset and predictions from G model, and train G simultaneously to minimize $\log(1 - D(G(X)))$ by:

$$\begin{aligned} \min_G \max_D V(D, G) = \mathbb{E}_{Y \in p_{\text{data}}(Y)}[\log D(X, Y)] \\ + \mathbb{E}_{\hat{Y} \in p_{\text{pred}}(\hat{Y})}[\log(1 - D(G(X, Y)))] \end{aligned} \tag{3}$$

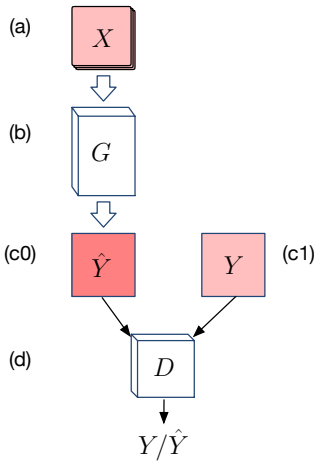

Figure 1: A baisc next frame prediction GAN, (a) input frames, (b) generative model used to to predict next frame (c0) predicted frame, (c1) ground truth frame (d) discriminative model to distinguish between groundtruth frame Y and predicted frame $\hat{Y}$.

Nevertheless, such a GAN is insufficient, as convolutions only account for short-range dependencies due to the limitation of kernel sizes in convolutional neural networks.

3.2 Multi-Scale CGANs

In order to tackle the problem of short-range dependencies in CNNs, our G model is defined as a multi-scale model that contains a set of convolutional networks $\{G_0, G_1, G_2, G_3\}$, each of which learns corre-

sponding optimal parameters from input frames at a different level of the image pyramid.

Suppose we have an input image size 32×32, the image pyramid is constructed by subsampling the image with a smaller image shape, by a factor of two along each coordinate. The resulting image is then subject to the same procedure and the cycle is repeated three times until we build an image pyramid of images size (4×4, 8×8, 16×16, 32×32) at different level. As illustrated in Figure 2, the entire multi-scale representation will look like a pyramid, with the original image at the bottom and each cycle's resulting smaller image stacked one atop the other.

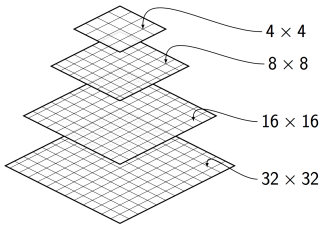

Figure 2: Image pyramid structure of input frame.

The image of lowest resolution at highest level now represents the global features and the one at the bottom preserves local information of the original image.

Let $S_1(4 \times 4), S_2(8 \times 8), S_3(16 \times 16), S_4(32 \times 32)$ be the size of the inputs of our multi-scale network. Let k be the scale level of the multi-scale network. Let u_k be the upscaling operator toward size S_{k+1}. Let X_k^i, Y_k^i denote the downscaled version of X^i and Y^i of size S_k, and G_k be a convolutional network that learns parameters to predict $Y_k - u_k(Y_{k-1})$ from X_k and a coarse guess of Y_k. We recursively define the network G_k, that makes a prediction $\hat{Y}_k$ of size S_k, by

$$\hat{Y}_k = u_{k-1}(\hat{Y}_{k-1}) + G_k(X_k, u_{k-1}(\hat{Y}_{k-1})) \qquad (4)$$

Generative model: Our model takes four consecutive frames as the input of generative networks. Then we scale them to different resolutions, which is 4×4, 8×8, 16×16 and 32×32 respectively. The generative model starts from the lowest scale $S_1(4 \times 4)$ by predicting a future frame $G_1(X_1)$ of scale S_1, given the input frames $\{X_1^1, X_1^2, X_1^3, X_1^4\}$ and then adding on the upsampled prediction $\hat{Y}_1$ of scale $S_2(8 \times 8)$ linearly to both the input and output of the higher resolution scale network G_2.

The generative model continuously makes a series of predictions using the prediction of size S_k as a starting point to make the prediction of size S_{k+1}. Eventually, the generative model G predicts 1 frame of 4 different scale versions $\{\hat{Y}_1, \hat{Y}_2, \hat{Y}_3, \hat{Y}_4\}$.

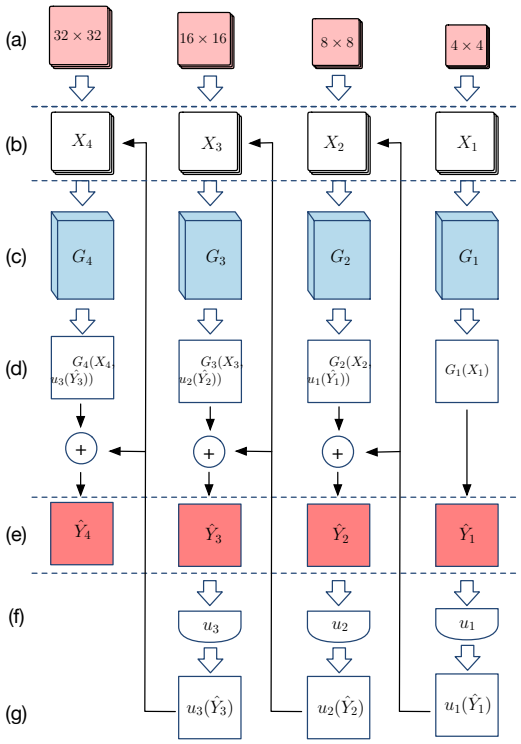

Figure 3: Our Multi-scale conditional convolutional generative models including following components: (a) input frames at a different level of the image pyramid, (b) the input layer, (c) convolutional neural networks, (d) the output layer, (e) predicted frames, (f) upscaling operators, (g) upscaled predicted frames.

The GAN is now extended as a multi-scale conditional generative adversarial net (CGAN), where each scale of model G, except the lowest scale network, receives an additional information as input, that is the upsampled predicted frame from lower-scale network.

Discriminative model: The discriminative model D is also a multi-scale convolutional network that consists of a set of convolutional networks $\{D_1, D_2, D_3, D_4\}$, each network D_k of which takes the generated frame as well as the ground truth frame as input, and learns to distinguish between predictions $\hat{Y}_k$ from generative model and corresponding ground truth frames Y_i from the dataset, with a single scalar output.

The value function of our extended adversarial model thus becomes:

$$\min_{G_k} \max_{D_k} V_k(D_k, G_k) = \mathbb{E}_{Y_k \in p_{\text{data}}(Y_k)}[\log D_k(X_k, Y_k)] + \mathbb{E}_{\hat{Y}_k \in p_{\text{pred}}(\hat{Y}_k)}[\log(1 - D_k(G_k(X_k, Y_k)))] \qquad (5)$$

3.3 Loss Functions

Training D: Let us take a sequence of n frames and m subsequent frames from the dataset, which can be noted as (X, Y) respectively. For each scale k, we perform one stochastic gradient descent (SGD) iteration of D_k while keeping the weights Θ_g of G fixed. The D is trained to assign a target label 1 to (X_k, Y_k), and to assign a target label 0 to $(X_k, G_k(X_k))$. Therefore, the loss function we use to train D is:

$$\ell^D_{\text{adv}}(X, Y, \Theta_d, \Theta_g) = \sum_{k=1}^{N} \ell_{\text{bce}}(D_k(X_k, Y_k), 1) + \ell_{\text{bce}}(D_k(X_k, G_k(Y_k)), 0) \tag{6}$$

where ℓ_{bce} is the binary cross-entropy loss, which is

$$\ell_{\text{bce}}(I, \hat{I}) = -\sum_i \hat{I}_i \log(I_i) + (1 - \hat{I}_i) \log(1 - I_i) \tag{7}$$

where I_i takes in $\{0, 1\}$, and $\hat{I}_i$ in $[0, 1]$.

Training G: Let us take a different sequence of n frames and m subsequent frames from the dataset, (X, Y). While keeping the weights Θ_d of D fixed, we perform 1 SGD step on G to minimize the adversarial loss:

$$\ell^G_{\text{adv}}(X, y, \Theta_d, \Theta_g) = \sum_{k=1}^{N} \ell_{\text{bce}}(D_k(X_k, G_k(X_k)), 1) \tag{8}$$

Training G adversarially minimizes the loss ℓ^G_{adv} and misleads the discriminative model D into believing that the prediction is from the dataset.

Combing Losses: However, minimizing this loss ℓ^G_{adv} alone can lead to instability, because G can always generate samples that misleads D. Therefore, we train the G to minimize a combined loss, $\lambda_{\text{adv}}\ell^G_{\text{adv}} + \lambda_2\ell_2$, that consists of the adversarial loss and of the ℓ_2 loss. There is therefore a tradeoff to adjust, by the mean of the λ_{adv} and λ_{ℓ_2} parameters, between sharp predictions due to the adversarial principle, and similarity with the ground truth brought by the second term.

Combining the loss functions, we can get the final loss:

$$L(X, Y) = \lambda_{\text{adv}}\ell_{\text{adv}} + \lambda_2\ell_2 \tag{9}$$

The training process is summerized in Algorithm 1:

3.4 Network Structure

The generative model G contains 4 scale networks, each network structure of which is shown in Table 1. Each layer contains padded convolutions interlaced with ReLU non-linearities. A Hyperbolic tangent(Tanh) is added at the end to constrain the output values into [-1,1].

The discriminative model D also consists of 4 scale networks that are described in Table 1 in network architecture details. Each layer uses standard non-padded convolutions followed by fully connected layers and ReLU non-linearities.

Algorithm 1 Multi-Scale CGANs for Physics Learning

1: Data preprocessing.
2: Define the networks structure for G and D models.
3: Initialization of weights, biases and variables.
4: **for** number of training steps **do**
5: **Update D:**
6: Input a batch of data samples $(X, Y) = (X^1, Y^1), ..., (X^N, Y^N)$ of size N.
7: $\Theta_d = \Theta_d - \alpha_d \sum_{i=1}^{N} \frac{\partial \ell^D_{\text{adv}}(X^{(i)}, Y^{(i)})}{\partial \Theta_d}$
8: **Update G:**
9: Input a another batch of data samples $(X, Y) = (X^1, Y^1), ..., (X^N, Y^N)$ of size N.
10: $\Theta_g = \Theta_g - \alpha_g \sum_{i=1}^{N} (\frac{\lambda_{\text{adv}} \partial \ell^G_{\text{adv}}(X^{(i)}, Y^{(i)})}{\partial \Theta_g} + \frac{\lambda_{\ell_2} \partial \ell_{\ell_2}(X^{(i)}, Y^{(i)})}{\partial \Theta_g})$
11: **end for**

Table 1: Architecture of Multi-scale CGANs

G **networks**	$G1$	$G2$
Feature maps	128,256,128	128,256,128
Kernel size	3,3,3,3	5,3,3,5
G **networks**	$G3$	$G4$
Feature maps	128,256,512,256,128	128,256,512,256,128
Kernel size	5,3,3,3,5	7,5,5,5,5,7
D **networks**	$D1$	$D2$
Feature maps	64	64,128,128
Kernel size	3	3,3,3
FC layer	512,256	1024,512
D **networks**	$D3$	$D4$
Feature maps	128,256,256	128,256,512,128
Kernel size	5,5,5	7,7,5,5
FC layer	1024,512	1024,512

4 Experiments

We evaluate the multi-scale CGANs using two different datasets, the tetherball dataset from TU Darmstadt Intelligent Autonomous Systems Labs (IAS) and the Physics101 dataset from MIT Computer Science and Artificial Intelligence Laboratory (CSAIL). The model was trained on a cluster of Competence Center of High Performance Computing in Hesse (HPC Hessen), with a GPU acceleration of NVIDIA Tesla K40M.

Tetherball dataset consists of three videos, in which a baseball is hung from a rope from a stationary metal pole and is hit counterclockwise. The ball keeps winding around the pole until it stops. Thus, we train our model to learn the swinging trajectory of the baseball. Ideally, the background including the white wall, the pole and the tables should remain still in predictions, and the predicted baseball is able to follow the trajectory of the ground truth baseball.

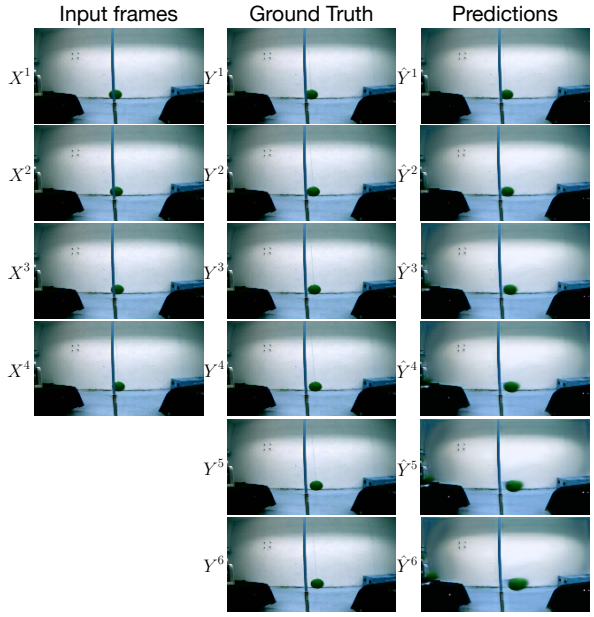

Figure 4: The result on Tetherball dataset. $\hat{Y}^1$ is the first predicted frame that corresponds to the ground truth frame Y^1. The first prediction was based on only 4 input frames $(X^1, ..., X^4)$. Later predictions $(\hat{Y}^2, ..., \hat{Y}^6)$ were computed recursively.

We train our network adversarially on the tetherball dataset with the suggested hyperparameters [1], using the open-source code [2]. As the result shown in Figure 4, in the predicted 6 frames, the first generated frame $\hat{Y}^1$ was predicted from 4 input frames(X^1, X^2, X^3, X^4), that come from the dataset, and the last frame $\hat{Y}^6$ was predicted from 4 previously generated frames $(\hat{Y}^2, \hat{Y}^3, \hat{Y}^4, \hat{Y}^5)$.

Given 4 concatenate frames, our model can successfully predict 7 frames of the still background and a short-range trajectory of the moving baseball of high sharpness.

Physics101 dataset consists of five different scenarios(Ramp, Spring, Fall, Liquid, Multi) of 101 objects made of different materials and with a variety of masses and volumes. We use the "Fall" videos, in which objects are held in the hand for a short period of time and then dropped in the air and freely fall onto various surfaces.

To demonstrate that our networks can generalize the physical features in videos, we train our networks adversarially using 139 videos of fall scenario of different objects, and test an unseen object on trained generative model.

In Figure 5, we compare two different scenarios (falling scenario and holding scenario).

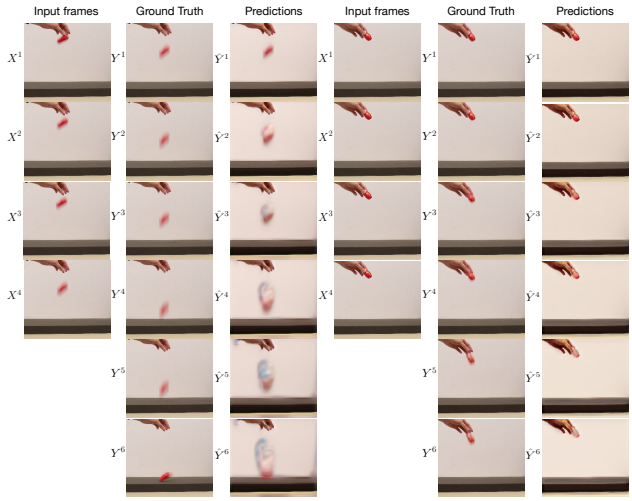

Figure 5: The results of Fall scenario on Physics101 dataset. Left: In the input frames, the test object is in the air. In the predictions, the object is falling down. Right: In the input frames, the test object is held in the hand. In the predictions, the object is still in the hand.

On the left side, the G model is tested with four concatenate input frames of the falling scenario, in which the object is falling in the air. The object is also falling towards ground in the prediction, thus our G is capable of predicting the future trajectory of falling object. Looking at only the last predicted frame $\hat{Y}^6$ and the last ground truth frame Y^6, in which the red part represents the object, we observe that the object reaches the same position on the surface in both frames. Namely, our model predicts the linear trajectory of the unseen object, with indirect representations of moving speed and directions.

On the right side, we test our model with four input frames of the holding scenario that the test object is held in the hand. In the predicted frames, the trajectory of the test object is a fixed position, however, the object starts falling in ground truth frame $\hat{Y}_3$. As a result, our G model fails to predict when the object will be dropped.

Training Losses: The training loss of the discriminative model is shown in Figure 6 and of the generative

model is in Figure 7.

The training loss of the discriminative model converges fast in about 3000 steps and ends up a low value on average with small variations. The training process of the generative model, however, is much more unstable. The G loss goes down on average but widely oscillates about the trend.

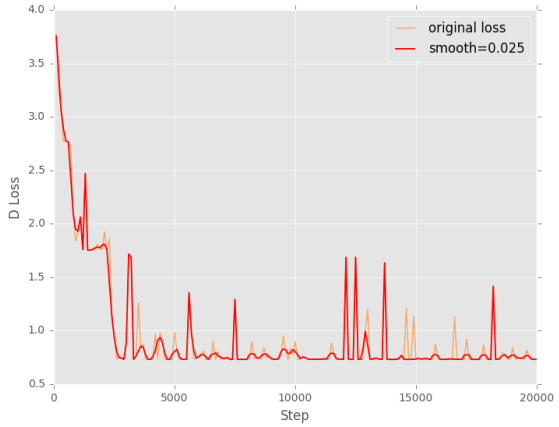

Figure 6: Training loss of discriminative model.

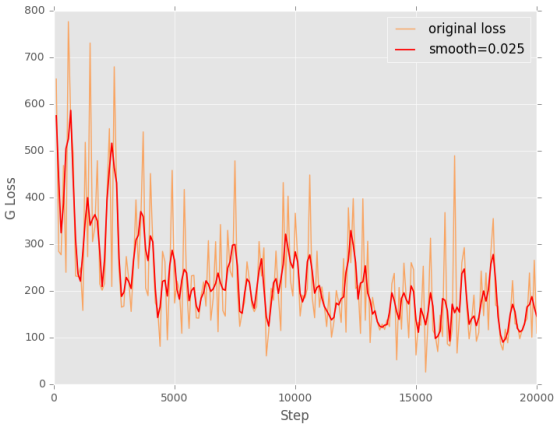

Figure 7: Training loss of generative model.

5 Conclusion

In this work, we tackle the problem of learning intuitive physics of objects from unlabeled videos and predicting the object's trajectory, using a multi-scale CGAN [1]. Our experiments show that, by training the generative model simultaneously with a discriminative model, the networks transform a set of high-dimensional video data into a set of feature representations. The trained generative model predicts plausible video sequences that the test object follows a short-range ground truth trajectory given that the object is moving in input frames, or remains in a position when prior input trajectory of the test object is a fixed point.

Using multi-scale CGANs to learn intuitive physics from videos has advantages and disadvantages. An advantage is that the learning is fully unsupervised, no need to label the videos manually. Additionally, our networks reduce high-dimensional pixel data into a set of feature representations that are capable of synthesizing the input videos and predicting plausible video frames of objects' trajectory.

The disadvantage is that the training process of the generative model is quite unstable, which may lead to failure representations. Another disadvantage is that our predictions are linear, namely, trajectories with sudden changes are hard to predict, e.g., our model fails to predict when will the object be dropped, and to which direction does the object bounce, after the object hits the surface.

Thus, an important direction of future work is to build a multi-scal GANs that learns non-linear physics representations by conditioning the network input on prior physical parameters, e.g., mass, forces, materials, etc.

Another application of GANs is to learn optimal policy directly from expert trajectories, a model-free imitation learning algorithm beyond behavioral cloning and inverse reinforcement learning that either requires large amounts of demonstration data, or is extremely expensive to run. For example, Jonathan Ho [15] proposed a generative adversarial framework for directly extracting a policy from data.

6 Acknowledgments

We would like to thank Simone Parisi for providing the tetherball dataset and MIT CSAIL for sharing their Physics101 dataset. We would also like to acknowledge Matt Cooper for the open-source code on GitHub.

References

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

[2] Cooper, M.: Adversarial Video Generation, https://github.com/dyelax/Adversarial_Video_Generation

[3] Baillargeon, R., 2004. *Infants' physical world.* Current directions in psychological science, 13(3), pp.89-94.

[4] Wu, J., Yildirim, I., Lim, J.J., Freeman, B. and Tenenbaum, J., 2015. *Galileo: Perceiving physical object properties by integrating a physics engine with deep learning.* Advances in neural information processing systems (pp. 127-135).

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

[6] Finn, C., Goodfellow, I. and Levine, S., 2016. *Unsupervised learning for physical interaction through video prediction.* Advances In Neural Information Processing Systems (pp. 64-72).

[7] Srivastava, N., Mansimov, E. and Salakhutdinov, R., 2015, March. *Unsupervised Learning of Video Representations using LSTMs.* ICML (pp. 843-852).

[8] Walker, J., Gupta, A. and Hebert, M., 2015. *Dense optical flow prediction from a static image.* Proceedings of the IEEE International Conference on Computer Vision (pp. 2443-2451).

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

[10] Huang, D.A. and Kitani, K.M., 2014, September. *Action-reaction: Forecasting the dynamics of human interaction.* European Conference on Computer Vision (pp. 489-504).

[11] Walker, J., Doersch, C., Gupta, A. and Hebert, M., 2016, October. *An uncertain future: Forecasting from static images using variational autoencoders.* European Conference on Computer Vision (pp. 835-851).

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

[13] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. *Generative adversarial nets.* Advances in neural information processing systems (pp. 2672-2680).

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

[15] Ho, J. and Ermon, S., 2016. *Generative adversarial imitation learning.* Advances in Neural Information Processing Systems (pp. 4565-4573).