# Learning the Intuitive Physics of Non-Rigid Object Deformations

**Zhihua Wang*** **Stefano Rosa*** **Andrew Markham**
Department of Computer Science, University of Oxford, United Kingdom
name.surname@cs.ox.ac.uk

## Abstract

The ability to interact and understand the environment is a fundamental prerequisite for a wide range of applications from robotics to augmented reality. In particular, predicting how deformable objects will react to applied forces in real time is a significant challenge. Objects in the real world are also often affected by occlusions, noise and missing regions. We present a framework, 3D-PhysNet, which is able to predict how a three-dimensional solid will deform under an applied force using intuitive physics modelling. In particular, we propose a new method to encode the physical properties of the material and the applied force, enabling generalisation over materials. The key is to combine deep variational autoencoders with adversarial training, conditioned on the applied force and the material properties. We further propose a cascaded architecture that takes a single 2.5D depth view of the object and predicts its deformation. Training data is provided by a physics simulator. The network is fast enough to be used in real-time applications from partial views. Preliminary experimental results show the generalisation properties of the architecture with simulated and real objects.

## 1 Introduction

Common-sense reasoning about intuitive physics of the scene, analogous to human intuition, is crucial for unconstrained interaction with the environment Wu *et al.* [2015, 2017]. In particular, the ability to understand the effect of applied forces on objects is vital for general purpose robotic applications. Traditionally, mobile navigation approaches consider all objects as static, rigid obstacles. Similarly, in robotic grasping applications, objects are often considered rigid and non-deformable at the perception level, with the resultant deformations only taken into account at the later control stage. However in the real world, many objects are non-rigid and change in shape/size when subjected to external forces. The ability to infer likely deformations is of great use for predictive control. Forward simulators based on Finite Element Models (FEM) are typically used to compute body deformations. Although they are highly accurate, a full mesh representation of a solid is required, and they are unable to work with incomplete depth views of an object, such as those available to a robot in the real world. Moreover, they are computationally expensive, unable to be used in real-time. As an alternative, we explore conditional deep models to learn the underlying physics of deformation. Deep generative networks have been applied with success to the problem of reconstructing 3D objects from partial views or synthesizing 3D objects Yang *et al.* [2017]; Han *et al.* [2017]. In particular, *Conditional Variational Autoencoders* (cVAEs) offer a natural way to encode the effects of physical properties and applied forces. Combining variational autoencoders with adversarial learning allows to complement the VAE reconstruction loss with the perceptual-level representation of the discriminator.

We propose a variational architecture that combines variational inference, a U-Net architecture and adversarial learning, trained on synthetic data from an FEM-based physics simulator. Given a single

*These two authors contributed equally

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

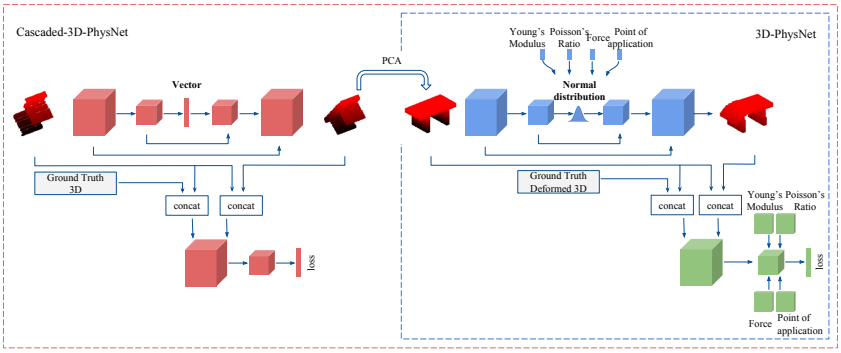



Figure 1: The 3D-PhysNet architecture. Blue blocks represent the generator network; green blocks represent the discriminator network. Red block represent the 3D object reconstruction module used in the cascaded 3D-PhysNet.

depth image of the deformable object and a conditioning input which includes the properties of the material, the strength of the force, and the location of the force, the network is able to output a predicted 3-D deformation of the solid. This prediction can then be used for tasks as diverse as robot manipulation and grasping, terrain deformation assessment, and in general for predicting the effect of forces on non-rigid objects in the context of end-to-end learning of intuitive physical models of the environment.

The intuition is that using variational inference it is possible for the network to learn the effect of the physical quantities that describe the elasticity and compression properties of the material. This enables the network to generalise over a wide range of materials, relaxing the need for a large training set. A single prediction is more than three orders of magnitudes faster that an equivalent FEM simulation, at the cost of lower resolution; this makes the approach useful for online evaluation for time-sensitive tasks.

## 2 Proposed Network Architecture

Figure 1 (right) shows the architecture of the proposed 3D-PhysNet. It is composed of a generator network $G$ and a discriminator network $D$, that are competing against each other.

Broadly, the generator maps the undistorted 3-D model into a deformed 3-D model, conditioned on the supplied parameters. The discriminator is used during training only and is a classifier that determines whether its input is drawn from the ground-truth or the output of the generator. The generator and discriminator are adversarial i.e. they each get better over time.

The generator is implemented as a variational autoencoder network and takes as input a voxel grid of size 64×64×64, representing a 3-D point cloud, which is obtained by voxelizing the input 2.5D depth image. To facilitate the replication of local structures and object details, the generator uses a U-Net structure with skip connections between encoder and decoder. The encoder E has five 3-D convolutional layers, followed by a fully-connected layer flattening the 3-D representation into a 1-dimensional vector, in turn followed by two layers $\mu$ and $\sigma$, representing the reparameterized mixture of gaussians from which we extract random samples. The condition vector encapsulates the material elasticity properties, the magnitude of the force, the location of the force (see Section **??**), and is concatenated with the latent vector.

The decoder follows the inverse of the encoder, with five deconvolutional layers followed by ReLU activations except for the last layer which is followed by a sigmoid function. The autoencoder reconstruction loss $\mathcal{L}_{ae}$ is a specialized form of Binary Cross-Entropy (BCE), as in Brock *et al.* [2016], and is given by: $\mathcal{L}_{ae} = -\alpha t \log(o) - (1-\alpha)(1-t)\log(1-o)$, where $t$ is the true binary value for each voxel (0,1), $o$ is the output value predicted by $E$ and is in the range (0,1), $\alpha$ is a parameter that balances false positives against false negatives.

The total generator loss is: $\mathcal{L}^g = \beta\mathcal{L}_{VAE} + (1-\beta)\mathcal{L}^g_{gan}$, where $\beta$ is a weight that balances the VAE loss and the GAN loss. Intuitively, the VAE loss guides the coarse 3D reconstruction of the object,

and is important in the first phase of training, while the GAN loss is useful for learning to generate more plausible predictions, in particular the subtle shape deformations caused by the condition vector.

The discriminator evaluates whether the predicted deformations from the generator are realistic, by classifying them as real or fake compared to the real input. Similar to the encoder, it is composed of five 3-D convolutional layers. The discriminator takes as input pairs of 'real' ground truth voxel grids and a 'fake' generated voxel grids from the generator, as well as the physics conditions. The condition vector is reshaped so that it can be concatenated with the voxel grid. Instead of outputting a binary value, the discriminator outputs a dense vector representing voxel similarities. The discriminator loss is based on WGAN-GP Gulrajani *et al.* [2017] which adopts Wasserstein Distance as a metric of similarity.

We also propose a cascaded framework, Cascaded-3D-PhysNet, in which the output of a 3D reconstruction network is fed into the input of 3D-PhysNet, as shown in Figure **??**. In this configuration we factor the learning into two separate, independently trained problems, the first performing 3-D reconstruction from a 2.5D point cloud and the second responsible for deforming the 3-D model. Our intuition is that the latent encoding provided by the GAN is more suited for learning high resolution reconstruction, subject to arbitrary rotation, whereas the normal encoding of the VAE is better suited for representing the physically-based smooth functions. We are agnostic to the reconstruction network used, in our approach we adopt 3D-RecGAN framework **?**. To simplify the task of 3D-PhysNet, PCA is to used to align the rotated shape along its principal axes.

**Encoding physical properties:** The deformation of a solid can be defined with the fuction $f_{def}$ : $\mathbf{x} \mapsto \mathbf{x} + \mathbf{d}$, where $\mathbf{x}$ is the set of points representing the undeformed solid and $\mathbf{d}$ represents a deformation field. Under the assumption that the body is isotropic (composed of a homogeneous material) and linearly elastic, the following relationship (an extension of Hooke's Law) relates applied force (stress) to resultant deformation (strain): $\boldsymbol{\sigma} = C\boldsymbol{\epsilon}$, where $\boldsymbol{\sigma}$ is the stress tensor, $\boldsymbol{\epsilon}$ is the strain tensor and $C$ is the Cauchy tensor mapping strain to stress. $C$ only depends on two physical parameters: the Young's modulus $E$ and Poisson's ratio $\nu$.

The Young's modulus describes the force needed to enlarge or compress a material by a fixed amount and is defined by the ratio of stress to strain in the direction of the applied force. In practice, it the stiffness (or elasticity) of a material. The Poisson's ratio denotes the negative ratio of the transverse strain over the axial strain. When a material is compressed in one direction, an expansion is observed in the other two perpendicular dimensions, and vice versa. In practice, it describes the compressibility of a material. Young's modulus is measured in GigaPascals (GPa) in the SI system and is in the range $(0, \infty)$. We fix an upper bound of 23, which corresponds to the elasticity of concrete. We sample $E$ logarithmically over the range and normalize to $[0, 1]$. The Poisson's ratio varies in the range $[0, 0.5]$. Rubber has a Poisson's ratio of 0.5 (perfect volume conservation), while most materials are in the range 0.25 to 0.48. We therefore sample $\nu$ linearly over the full range.

## 3 Experimental Results

The network was implemented with Tensorflow 1.4 [2] and trained on a single Nvidia Pascal Titan X GPU, and trained with a batch size of 8 using the Adam optimizer, with $lr = 5e-5$, $\beta 1 = 0.5$, $\beta 2 = 0.999$,$\epsilon = 1e-8$. The prediction time for a single input is 35.7ms, up to three orders of magnitude faster than an equivalent FEM simulation and allowing for real-time deformation estimation. In the cascaded configuration we use 3D-RecGAN Yang *et al.* [2017] for shape reconstruction from rotated partial viewpoints. The network has a comparable prediction time.

In this work we used the COMSOL Multiphysics software in order to generate the training voxel grids pairs and relative condition vectors, but the network is agnostic to the simulator. The one-dimensional condition vector is obtained by concatenating the object's Young's modulus, the Poisson's ratio, the location of the force and its magnitude. We vary the Young's modulus and the Poisson's ratio over their whole range, using different sampling approaches. We vary the force magnitude over 30 values and the point of application of the force over 10 positions along the top of the object. We also generate a second dataset in which we stretch the object along the x-y-z axes, over 8 scales.

**Generalisation over physical parameters and scales:** In this experiment we analyse the ability of the network to generalise over the various conditions, as well as scale variations. We train 3D-PhysNet

[2] `https://github.com/vividda/3D-PhysNet`

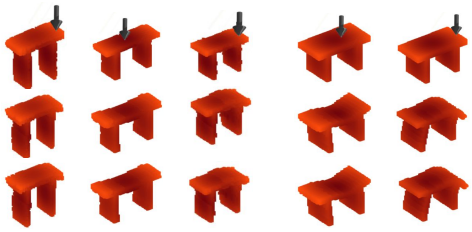
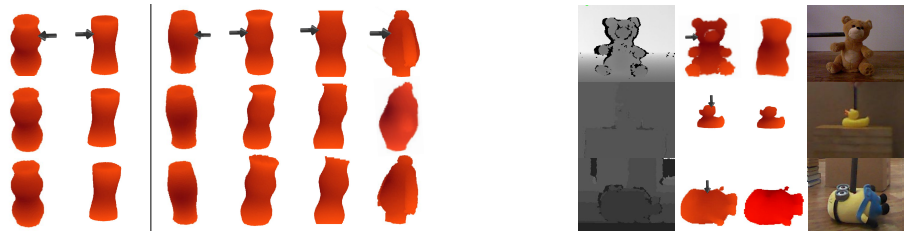


Figure 2: Generalisation over shape variations and applied forces. Top row: undeformed object and applied force; middle: predicted deformations; bottom row: ground truth.



(a) (b)

Figure 3: Examples of predicted deformations for different objects. (a) Cross-category predictions on unseen objects; top: undeformed object and applied force; middle: predicted deformation; bottom: ground truth. The three columns on the right were only trained on a set of cylinder-like objects. (b) Cross-category prediction on a real objects; left: depth input image; middle: input voxel and predicted deformation; right: actual deformation.

with full 3D inputs and vary both $E$ and $\nu$ over 20 values each. We then test on unseen values of $E$ and $\nu$. We also generate variations of the input shape by stretching it along the three axes. The network converges after 14k iterations and has a resultant IOU of 0.98.

**Multi-category and cross-category prediction:** Finally, to further investigate the generality of our network we train on a set of cylinder-like objects (first and second column of Figure 3a) and then test on completely unseen objects (last four columns of Figure 3a). We also show some preliminary results from real depth images in Figure 3b. For real objects, a single depth view of the real objects is obtained using a Microsoft Kinect camera. The results show how the network is able to learn from simple primitives and generalise to unseen objects, both synthetic and from the real world. Note how, albeit simplified in its shape, the predicted shapes of the toys are being pushed by the force (the feet are fixed to the ground) and a simulacrum of the arms is present in the predicted shapes.

## References


Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Generative and discriminative voxel modeling with convolutional neural networks. *arXiv preprint arXiv:1608.04236*, 2016.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5769–5779, 2017.

X. Han, Z. Li, H. Huang, E. Kalogerakis, and Y. Yu. High-resolution shape completion using deep neural networks for global structure and local geometry inference. In *IEEE International Conference on Computer Vision (ICCV)*, October 2017.

Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In *Advances in neural information processing systems*, pages 127–135, 2015.

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 30*, pages 152–163. 2017.

Bo Yang, Hongkai Wen, Sen Wang, Ronald Clark, Andrew Markham, and Niki Trigoni. 3d object reconstruction from a single depth view with adversarial learning. In *2017 IEEE International Conference on Computer Vision (ICCV) Workshop*, 2017.