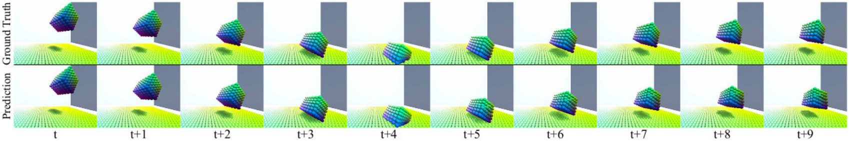
Flexible Neural Representation for Physics Prediction

Damian Mrowca, Chengxu Zhuang, Elias Wang, Nick Haber, Li Fei-Fei, Joshua B. Tenenbaum and Daniel L. K. Yamins

Presented by: Bebart M. Janbek
Formerly Department of Mathematics
Simon Fraser University

July 16, 2018

Capturing the World Physics



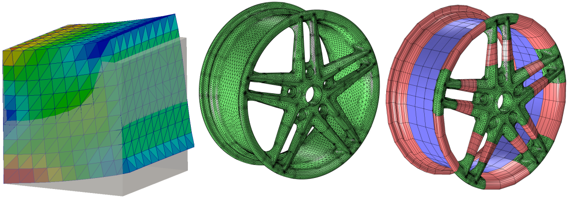
We are trying to capture

- Translations
- Rotations
- Deformations
- Collisions
- Object permanence
- Multiple Scales
- Occlusion

Capturing the World Physics

- Humans can make a reasonable prediction about the environment and objects interactions around them
- It is thought that they have their own physics rules that help them
- Learn about object permanence, occlusion, and deformability
- Replicating these capabilities in machines should be useful
- Prediction form images alone is not easy
- Traditional physics engines are hard wired and can not learn from their environment
- This paper proposes an end to end differentiable neural network to predict physics

Tools for Physics Simulation



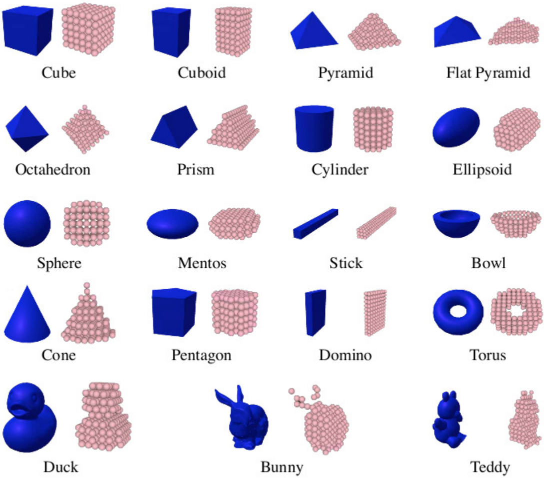
Mathematically accurate methods:

- Solid mechanics and fluid mechanics solvers (COMSOL).
- Partial differential equations (captures internal physics (stress/strain), boundary conditions, material properties or parametrization (stiffness)
- finite elements (numerical methods)

Problems:

- Slow and consume a lot of resources
- So faster physics based realistic simulators (FleX)

Points and Meshes

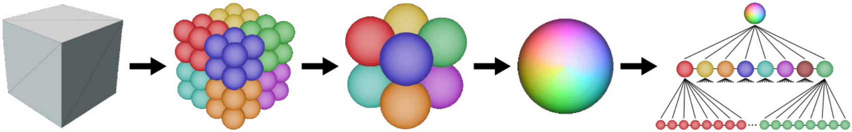


Forces and Effects

Effects in the paper are (most probably) intermediate representations of the action of the following forces on the object particles

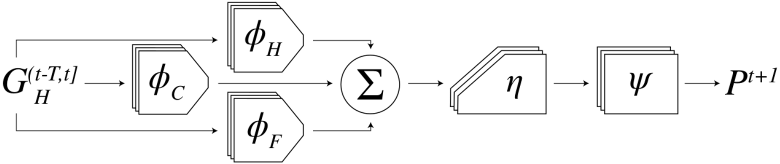
- Internal forces (element on element, or particle on particle)
- External force: body forces (gravity) or surface forces (collision)
- Unknown and unaccounted for forces.

Algorithm: Particles Hierarchy



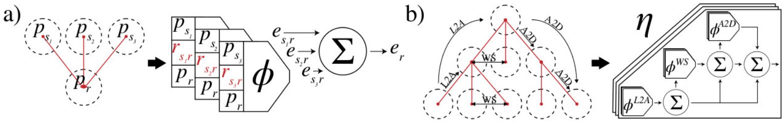
- Objects (root nodes) are divided into particles (graph leaves)
- Particles have labels $p_i = (x, \delta, m)$=(position,velociyy,mass)
- Direct edge label between p_i and p_j is a scalar r_{ij} (local stiffness) which is a learned parameter
- Sub-groupings of leaves are connected to intermediate nodes
- Leaves belonging to different intermediate nodes are disconnected.
- Intermediate nodes with neighboring leaves are connected
- Intermediate nodes labels are assumed to be centers of mass

Prediction



- Input graphs and states for previous time steps
- ϕ_F calculates the effects of external forces
- ϕ_C calculates the effects of collisions
- ϕ_H calculates the effects of history
- η propagates effects through internal interactions
- ψ gives a prediction of the new state P^{t+1}

Prediction: Internal Effects Propagation



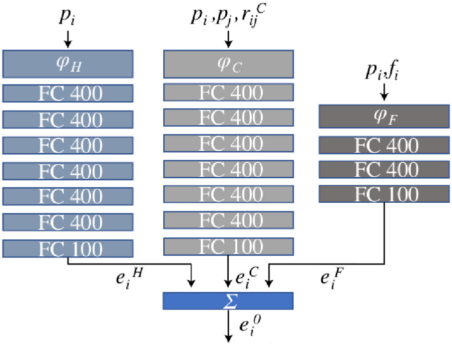
- Sender nodes p_s affect receiver nodes p_r through relations $r_{s,r}$
- Leaves affect ancestors, siblings affect each other, then ancestors affect descendants.

$$e_a^{L2A} = \sum_{p_l \in leaves(p_a)} \phi^{L2A}(p_l, p_a, r_{la}, e_l^0)$$

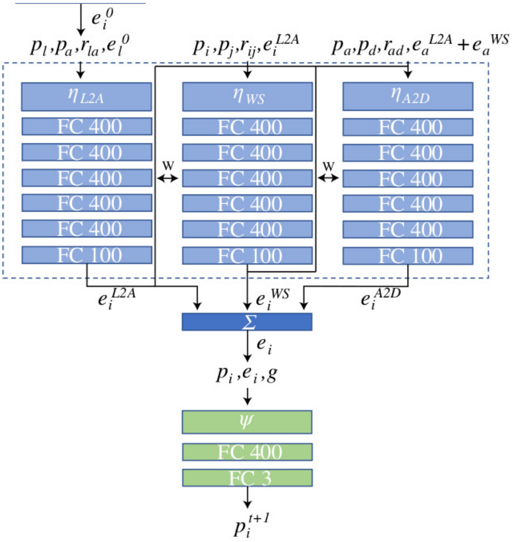
$$e_j^{WS} = \sum_{p_i \in sib(p_j)} \phi^{WS}(p_i, p_j, r_{ij}, e_i^{L2A})$$

$$e_d^{A2D} = \sum_{p_a \in anc(p_d)} \phi^{A2D}(p_a, p_d, r_{ad}, e_a^{L2A} + e_a^{WS})$$

Prediction Neural Network I



Prediction Neural Network II



Loss Function

$$\text{Loss} = \alpha \left(\sum_{p_i} \left| \hat{\delta}_{i,\ell}^{t+1} - \delta_{i,\ell}^{t+1} \right|^2 + \beta \sum_{p_i} \left| \hat{\delta}_{i}^{t+1} - \delta_{i}^{t+1} \right|^2 \right) + (1-\alpha) \sum_{p_i \in sib(p_j)} \left| \hat{d}^{t+1}(i,j) - d^{t+1}(i,j) \right|^2$$

- $\hat{\delta}$ and δ are position changes for prediction and ground truth respectively
- First term is normalized local loss (local parent coordinates)
- Second term is global loss

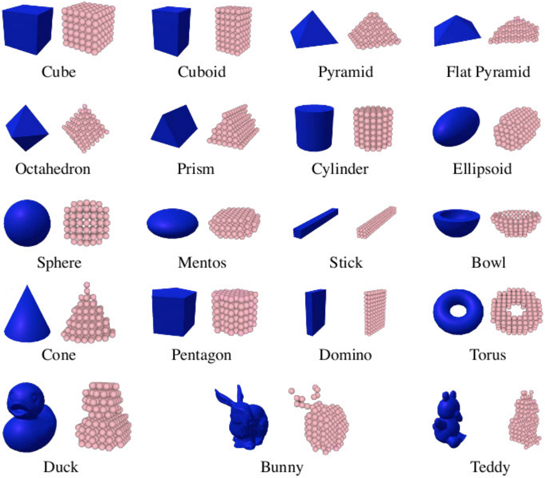
$$\delta_i^{t+1} = \delta_{i,\ell}^{t+1} + \sum_{j \in anc(i)} \delta_{j,\ell}^{t+1}$$

- Third term is pairwise distance loss

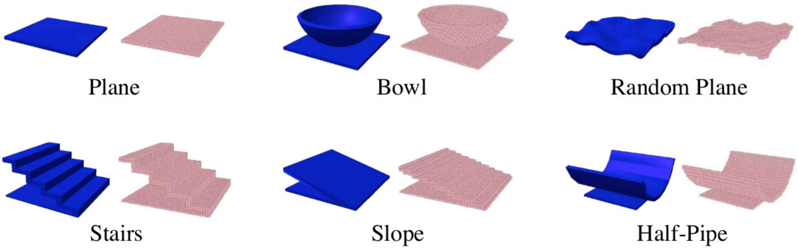
Training and Testing

- Build particle-based environment on top of FleX physics engine in Unity3D
- Particle representation created from 3D object mesh
- Randomly generate physics scenes for training
- Training done on 10 different rigid shapes
- Testing done on same object instances but with varying external forces
- Extended to complex shapes and deformable bodies
- Extended to shapes not seen during training
- Extended to interaction with complex surfaces
- Collision between moving objects
- Multi-object interaction
- Test with varying gravity, stiffness and mass

Experiments Objects



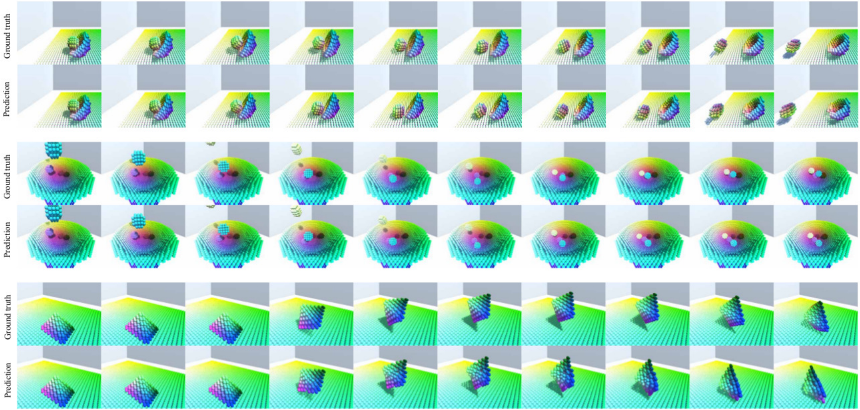
Experiments Surfaces



Experiments include

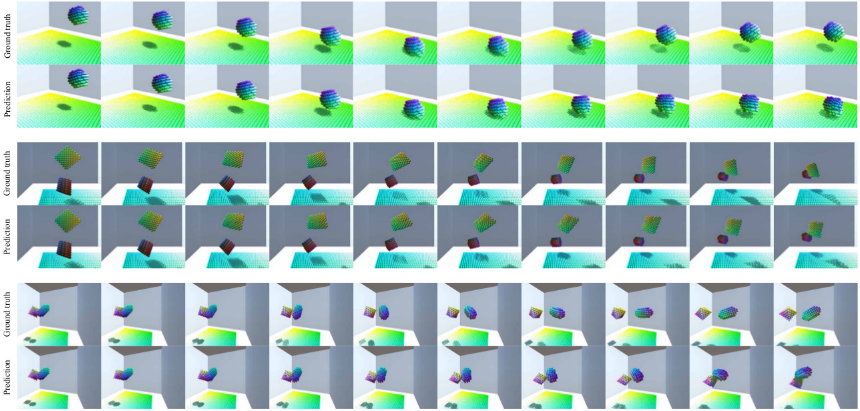
- Objects only interacting with surfaces through gravity
- Objects interacting with each other without gravity
- Forces applied at random
- Combinations and variations of the previous scenarios

Qualitative Results



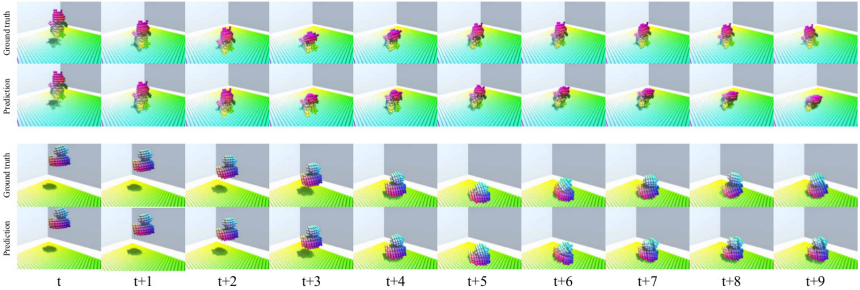
Sphere falling out of a bowl, spheres falling into bowl, and rigid pyramid colliding with floor.

Qualitative Results



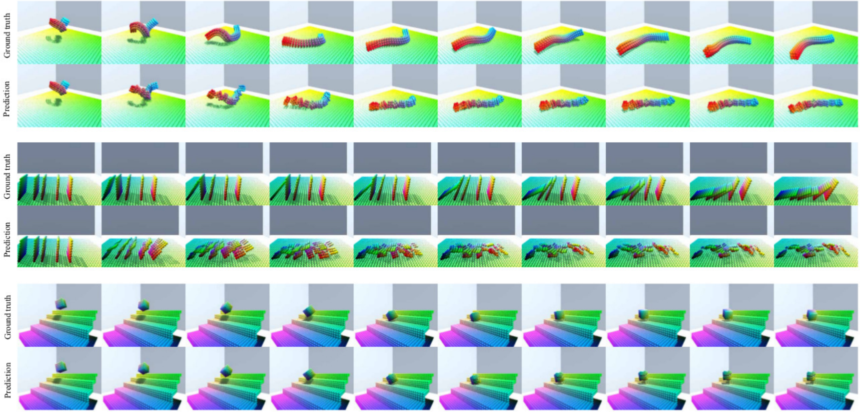
Rigid sphere colliding with floor, cylider colliding with pyramid and ellipsoid colliding with octahedron

Qualitative Results



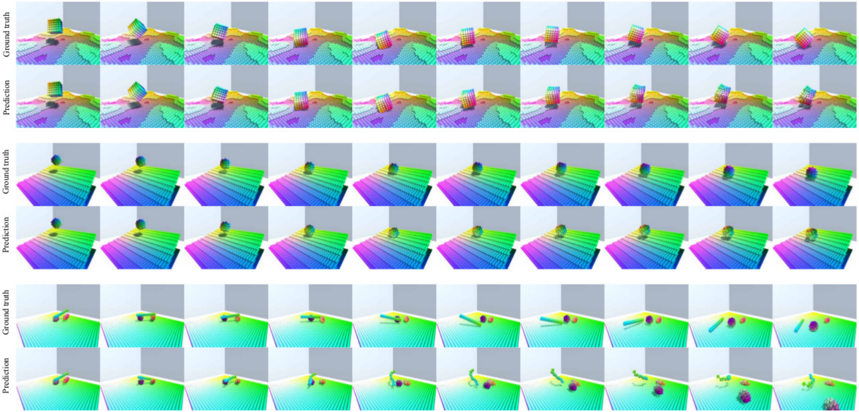
Soft teddy colliding with floor and soft duck colliding with floor

Qualitative Results



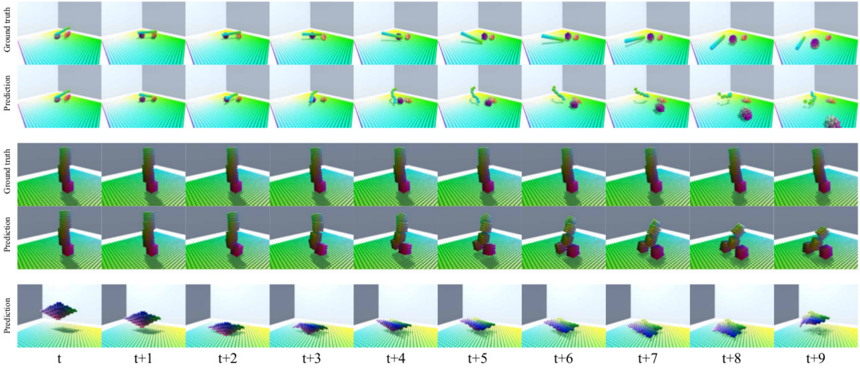
Very deformable stick, falling dominoes and rigid cube colliding with stairs

Qualitative Results



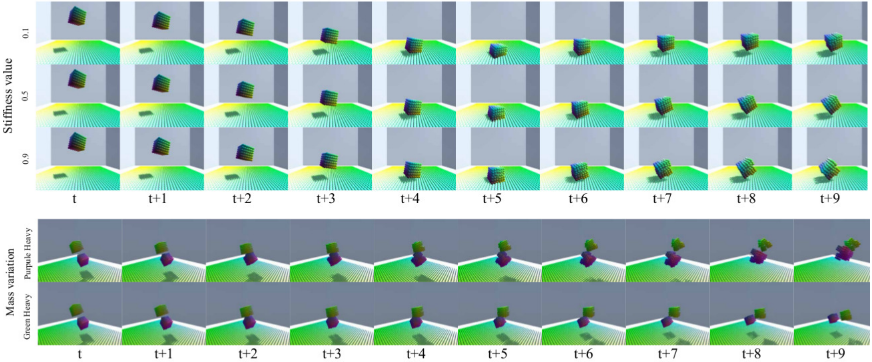
Cube colliding with random surface, ball on a slope and three objects colliding with each other.

Qualitative Results



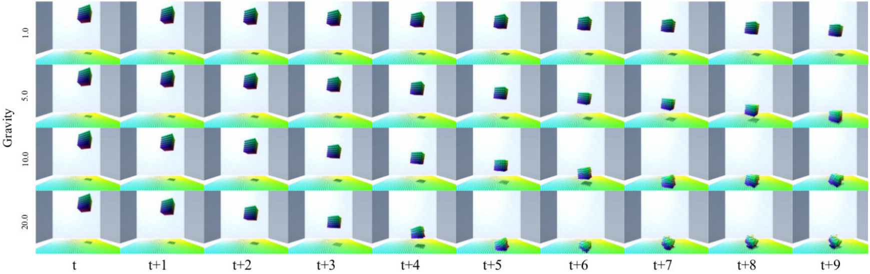
Three objects colliding, slowly falling tower and half-soft half-rigid object.

Qualitative Results: Stiffness and Mass



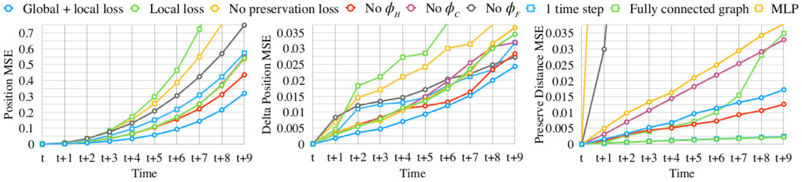
- Stiffness increase means less deformation
- Mass increase means less displacement after collision

Qualitative Results: Gravity



Larger gravity means faster descent.

Quantitative Results



- MLP: multilayer perceptron
- Fully-connected: no graph hierarchy
- No ϕ_F and ϕ_C: forces aggregated to input and fed directly into η