Learning Particle Dynamics for Manipulating Rigid Bodies, Deformable Objects, and Fluids

Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B. Tenenbaum, and Antonio Torralba
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
{liyunzhu,jiajunwu,russt,jbt,torralba}@mit.edu

Abstract

Real-life control tasks involve matter of various substances—rigid or soft bodies, liquid, gas—each with distinct physical behaviors. This poses challenges to traditional rigid-body physics engines. Particle-based simulators have been developed to model the dynamics of these complex scenes; however, relying on approximation techniques, their simulation often deviates from real world physics, especially in the long term. In this paper, we propose to learn a particle-based simulator for complex control tasks. Combining learning with particle-based systems brings in two major benefits: first, the learned simulator, just like other particle-based systems, acts widely on objects of different materials; second, the particle-based representation poses strong inductive bias for learning: particles of the same type have the same dynamics within. This enables the model to quickly adapt to new environments of unknown dynamics within a few observations. Using the learned simulator, robots have achieved success in complex manipulation tasks, such as manipulating fluids and deformable foam. The effectiveness of our method has also been demonstrated in the real world. Our study helps lay the foundation for robot learning of dynamic scenes with particle-based representations.

1 Introduction

We propose to learn a differentiable, particle-based simulator for complex control tasks, drawing inspiration from recent development in differentiable physical engines [Battaglia et al., 2016, Chang et al., 2017]. In robotics, the use of differentiable simulators, together with continuous and symbolic optimization algorithms, has enabled planning for increasingly complex whole body motions with multi-contact and multi-object interactions [Toussaint et al., 2018]. Yet these approaches have only tackled local interactions of rigid bodies. We develop dynamic particle interaction networks (DPI-Nets) for learning particle dynamics, focusing on capturing the dynamic, hierarchical, and long-range interactions of particles (Figure 1a).

Learning a particle-based simulator brings in two major benefits. First, the learned simulator, just like other particle-based systems, acts widely on objects of different materials. DPI-Nets have successfully captured the complex behaviors of deformable objects like fluids, rigid-bodies, and plasticine (Figure 1b). Second, the particle-based representation poses strong inductive bias for learning: particles of the same type have the same dynamics within. This enables the model to quickly adapt to new environments of unknown dynamics within a few observations.

Experiments demonstrate that DPI-Nets work well, significantly outperforming interaction networks [Battaglia et al., 2016] and a few other baselines. When applied in downstream control tasks, our model helps to complete complex manipulation tasks for deformable objects and fluids. It also adapts to cases with unknown physical parameters that need to be identified online. We have also performed real-world experiments to demonstrate our model's generalization ability.

Our project page: http://dpi.csail.mit.edu/

Preprint. Work in progress.

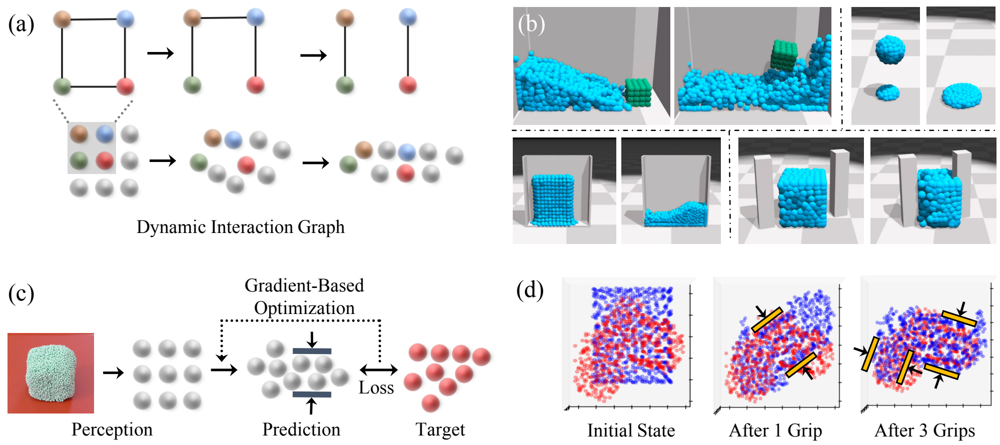


Figure 1: **Learning Particle Dynamics for Control.** (a) DPI-Nets learn particle interaction while dynamically building the interaction graph over time. (b) Simulation using learned DPI-Nets for rigid bodies, deformable objects, and fluids. (c) Perception and control with the learned model. Our system first reconstructs the particle-based shape from visual observation. It then uses gradient-based trajectory optimization to search for the actions that produce the most desired output. (d) Control with learned DPI-Nets to manipulate a deformable foam (blue) into a target shape (red).

2 Approach

2.1 Dynamic Particle Interaction Network

Particle-based system is widely used in physical simulation due to its flexibility in modeling various types of objects [Macklin et al., 2014]. We extend existing systems that model object-level interactions [Battaglia et al., 2016] to allow particle-level deformation, where each object is represented as a set of particles. We now define the graph on the particles and the rules for influence propagation.

Dynamic graph building. The vertices of the graph are the union of particles for all objects. The edges between these vertices are dynamically generated over time to ensure efficiency and effectiveness. The construction of the relations is specific to environment and task, and a common choice is to consider the neighbors within a predefined distance.

Hierarchical modeling for long-range dependence. Propagation networks Li et al. [2018] require a large number of propagation steps to handle long-range dependence, which is both inefficient and hard to train. Hence, we add one level of hierarchy to efficiently propagate the long-range influence among particles [Mrowca et al., 2018], where we cluster the particles into several non-overlapping clusters, and employ a multi-stage propagation paradigm.

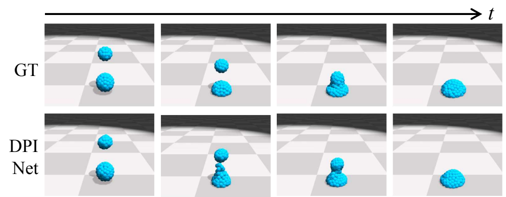
Applying to objects of various materials. *Rigid bodies:* All the particles in a rigid body are globally coupled; hence for each rigid object, we define a hierarchical model to propagate the effects, and predict a rigid transformation. *Elastic/Plastic objects:* For elastically deforming particles, only using the current position and velocity as the state is not sufficient, as it is not clear where the particle will be restored after the deformation. Hence, we include the particle state with the resting position to indicate the place to restore. *Fluids:* For fluid simulation, one has to enforce density and incompressibility. We build edges dynamically, connecting a fluid particle to its neighboring particles. The strong inductive bias leveraged in the fluid particles allows good performance even when tested on data outside training distributions.

For the interaction between different materials, two directed edges are generated for any pairs of particles that are closer than a certain distance.

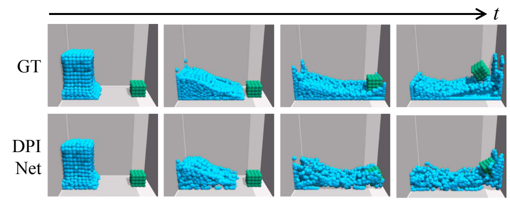
2.2 Control on the Learned Dynamics

Model-based methods offer many advantages when comparing with their model-free counterparts, such as generalization and sample efficiency. We can rollout using the learned dynamics and optimize the control inputs by minimizing a loss between the simulated results and a target configuration.

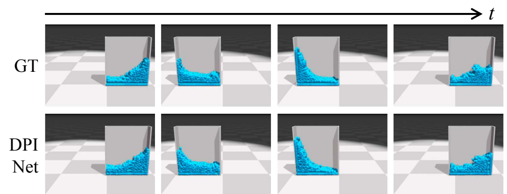
Model predictive control using shooting methods. The learned model might deviate from the reality due to accumulated prediction errors. We use Model-Predictive Control (MPC) [Camacho and Alba, 2013] to stabilize the trajectory by doing forward simulation and updating the control inputs at every time step to compensate the simulation error.



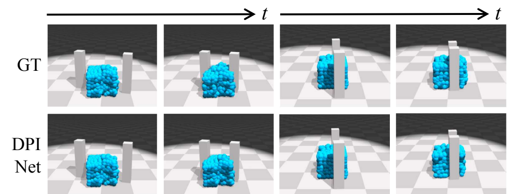
(a) FluidFall



(b) BoxBath



(c) FluidShake

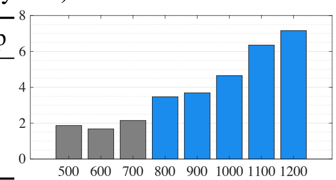


(d) RiceGrip

Figure 2: **Qualitative results on simulation.** We show side-by-side comparisons between the ground truth (GT) and the rollout from our model (DPI-Net) in four difference environments. (a) FluidFall - Two drops of high-viscosity fluids are falling down and merging with each other. (b) BoxBath - A block of water is flushing a rigid cube. (c) FluidShake - Shaking a box of fluids. (d) RiceGrip - Grip an object that can deform both elastically and plastically (e.g., sticky rice).

Methods	FuildFall	BoxBath	FluidShake	RiceGrip
IN [Battaglia et al., 2016]	2.74	N/A	N/A	N/A
DPI-Net w/o dynamic graphs	0.42	123.5	97.7	0.64
DPI-Net w/o hierarchy	**0.15**	3.40	**1.89**	0.29
DPI-Net	**0.15**	**2.58**	**1.89**	**0.13**

(a) Rollout discrepancy



(b) Generalization in FluidShake

Figure 3: **Quantitative results on simulation.** The errors showed here are the MSE ($\times 10^{-2}$) between the ground truth and model rollout. (a) FluidFall and FluidShake involve no hierarchy, so DPI-Net performs the same as the variant without hierarchy. Fixing the graph (i.e., w/o dynamic graphs) severely damages the modeling of flowing fluids as indicated in BoxBath and FluidShake. (b) The gray bars denote the range of particle numbers that have been seen during training, which indicate interpolation performance. The blue bars indicate extrapolation performance that our model can generalize to cases containing two times more particles than cases it has been trained on.

Online adaptation. In many real-world cases, without actually interacting with the environment, inherent attributes such as mass, stiffness, and viscosity are not directly observable. DPI-Nets can estimate these attributes on the fly with SGD updates by minimizing the distance between the predicted future states and the actual future states.

3 Experiments

Environments. *FluidFall* (Figure 2a) Two drops of fluids are falling down, colliding, and merging. *BoxBath* (Figure 2b) A block of fluids are flushing a rigid cube. In this environment, we have to model two different materials and the interactions between them. *FluidShake* (Figure 2c) We have a box of fluids and the box is moving horizontally. *RiceGrip* (Figure 2d) We manipulate an object with both elastic and plastic deformation (e.g., sticky rice). We use two cuboids to mimic the fingers of a parallel gripper, where the model has to learn the interactions between the gripper and the "sticky rice", as well as the interactions within the "rice" itself.

Results. Qualitative and quantitative results on physical simulation are in Figure 2 and Figure 3. We compare our method (DPI-Net) with three baselines, Interaction Networks [Battaglia et al., 2016], DPI-Net without hierarchy, and DPI-Net without dynamic graphs.

Figure 4 and Figure 5 show qualitative and quantitative results on control, respectively, where we conduct ablation studies on our model and compare with Model-free Deep Reinforcement Learning (Actor-Critic method optimized with PPO (Schulman et al. [2017])).

References

Peter W. Battaglia, Razvan Pascanu, Matthew Lai, Danilo Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. In *NIPS*, 2016.

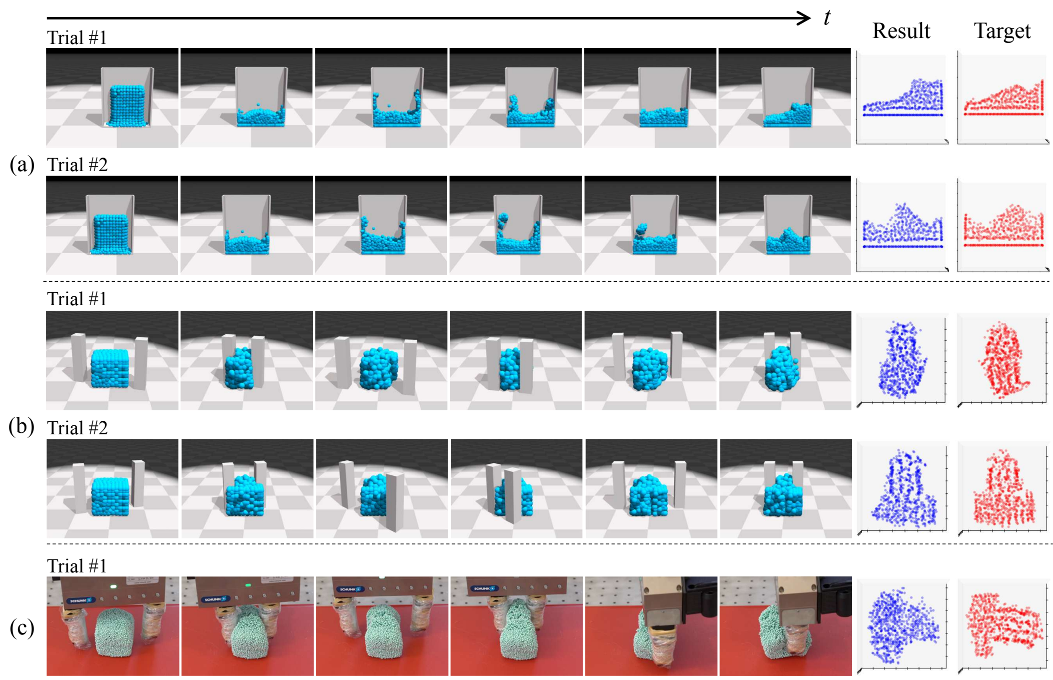
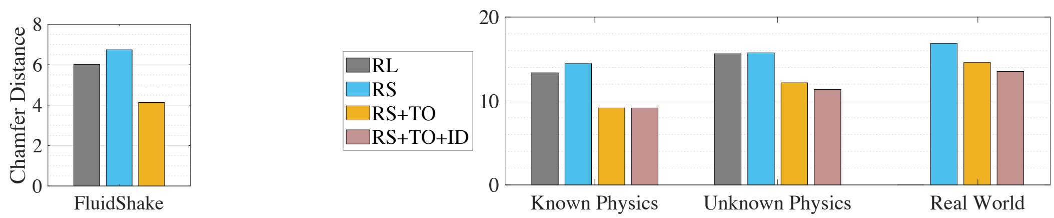


Figure 4: **Qualitative results on control.** (a) FluidShake - Manipulating a box of fluids to match a target shape. The Result and Target indicate the fluid shape when viewed from the cutaway view. (b) RiceGrip - Gripping a deformable object and molding it to a target shape. The Result and Target indicate the shape when viewed from the top. (c) RiceGrip in Real World - Generalize the learned dynamics and the control algorithms to the real world by doing online adaptation.



(a) (b) RiceGrip

Figure 5: **Quantitative results on control.** We show the control performance (as evaluated by the Chamfer distance ($\times 10^{-2}$) between the manipulated result and the target) for (a) FluidShake and (b) RiceGrip by comparing with baselines: RL - Model-free deep reinforcement learning optimized with PPO. RS - Random search the actions from the learned model and select the best one to execute. TO - Trajectory optimization augmented with model predictive control. ID - Online system identification by estimating uncertain physical parameters during run time.

Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer Science & Business Media, 2013.

Michael B Chang, Tomer Ullman, Antonio Torralba, and Joshua B Tenenbaum. A compositional object-based approach to learning physical dynamics. In *ICLR*, 2017.

Yunzhu Li, Jiajun Wu, Jun-Yan Zhu, Joshua B Tenenbaum, Antonio Torralba, and Russ Tedrake. Propagation networks for model-based control under partial observation. *arXiv preprint arXiv:1809.11169*, 2018.

Miles Macklin, Matthias Müller, Nuttapong Chentanez, and Tae-Yong Kim. Unified particle physics for real-time applications. *ACM TOG*, 33(4):153, 2014.

Damian Mrowca, Chengxu Zhuang, Elias Wang, Nick Haber, Li Fei-Fei, Joshua B Tenenbaum, and Daniel LK Yamins. Flexible neural representation for physics prediction. *arXiv preprint arXiv:1806.08047*, 2018.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv:1707.06347*, 2017.

Marc Toussaint, K Allen, K Smith, and J Tenenbaum. Differentiable physics and stable modes for tool-use and manipulation planning. In *RSS*, 2018.