Paper Review: Learning to Simulate Complex Physics with Graph Networks

Zhu Haoqiang

April,2023

Task and Contribution

This paper introduces a method to simulate complex physical systems using graph networks. The author proposes a new framework, named "Graph Network-based Simulators" (GNS), in which a particle is expressed as a node in a graph and the dynamics is learned by message-passing. The results show that the GNS framework advanced the state-of-the-art in learned physical simulation at that time.

Datasets

The datasets used in the paper are created by the authors themselves. They use Taichi-MPM engine to simulate most of the domain, such as water, sand, and goop. At the same time, they create a high-resolution 3D water scenario with a SPH-based fluid simulator and a BOXBATH secnario using PBD engine FleX. Most of the dataset contain 1000 train, 100 validation and 100 test trajectories. Each has 300 to 2000 timesteps.

Method and Experiments

In the preprocessing phase, a graph is constructed based on the position of particles. The nodes of the graph contain information about particle type, velocity in the past C frames, and displacement to the boundary. The edges contain relative position information between nodes. There are three main components of the model: an encoder, a processor, and a decoder. The encoder is an MLP that can learn the representations of nodes and edges. The processor is a stack of N graph networks with an identical structure. In the processor, MLPs are used as the update function to perform message passing in the graph. For the decoder, it is simply an MLP used to predict acceleration based on the node features output by the processor. To overcome the accumulation error problem in the rollout phase, the author injects noise into the training dataset.

The experiments show that the main determinants of the method are the number of message-passing steps and the connectivity radius of the graph. The results demonstrate that graph network works very effectively for complex physics tasks. At the same time, noise injection can effectively reduce rollout error.

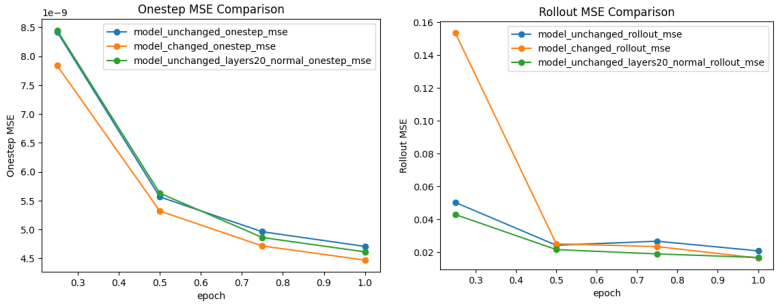
Limitation

In the FLUIDSHAKE-BOX dataset, the container is vigorously shaken, and the model fails to keep the shape of the rigid block. It also cannot learn static friction and adhesion well.

Improvements and Insight

I try several possible methods to improve the result, including changing the structure of the graph network layer, predicting vector's direction and norm separately, predicting the whole sequence of position instead of just one step acceleration, etc. Among all of them, changing the structure of the GN layer improves the result a little bit. I compare it with the result of increasing the number of GN layers from 10 to 20, and the original method. The result is shown below. We can see that in both one-step MSE and rollout MSE, It shows better performance than the original method. At the same time, it achieves better one-step MSE and rollout MSE with $\frac{1}{3}$ less parameter compare to the 20 GN layers model. Note that because of the limitation of GPU resource, I only train one epoch and repeat each experiment once. So the result need to be verified with more experiments.

The change I made is simple, the original GN layer update edge features by $mlp(node_i, node_j, edge) + edge$. I change it to $mlp(node_i+, node_j+, edge) + edge$, where $node_i+$ and $node_j+$ is the updated node features, I think it would work better because the edge features are not only depend on the its nodes, but also the neighbours of its nodes. By using the updated node features, its neighbours' effect is considered during the edge update stage.



Conclusion

The GNN-based physics simulation method proposed in this paper is an effective method with a wide range of applications. The method is able to learn complex physical laws and interactions between different materials, and has good accuracy and generalization.