Learning Mesh-Based Simulation with Graph Networks

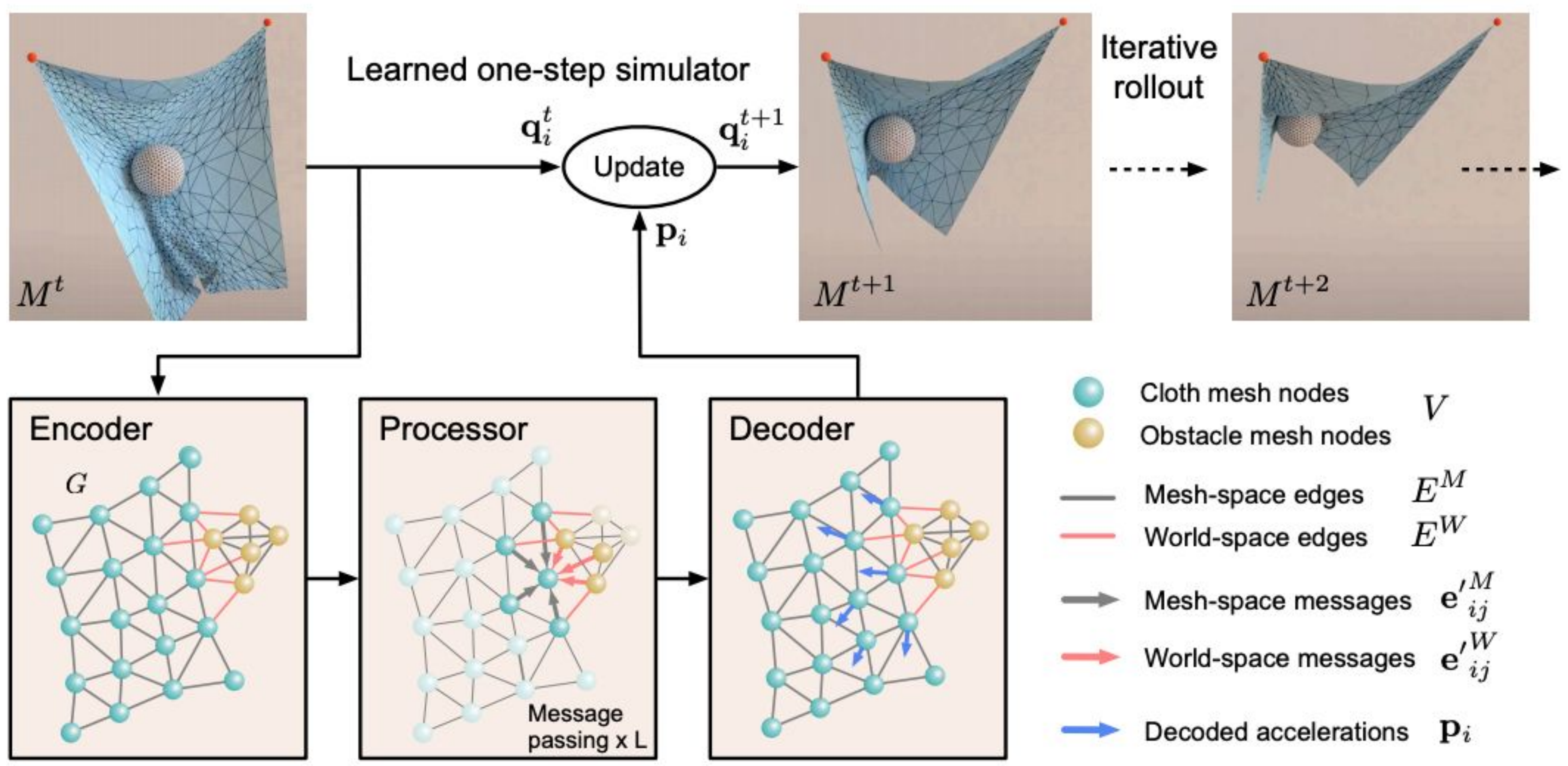
Tobias Pfaff*, Meire Fortunato*, Alvaro Sanchez-Gonzalez*, Peter Battaglia
DeepMind

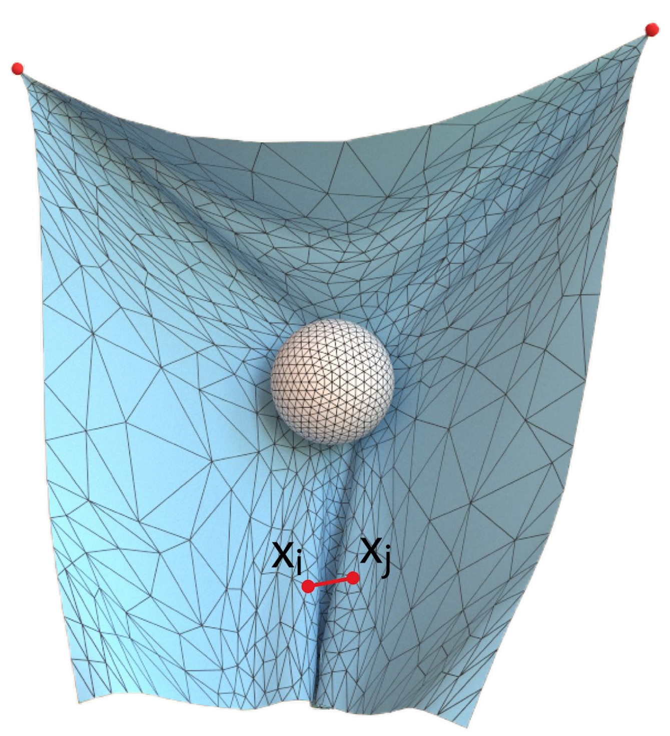
Motivation

Mesh-based simulations are central to modeling complex physical systems in many disciplines across science and engineering, yet they are underexplored in ML. Here we introduce MeshGraphNets, a graph neural network-based method for learning simulations, which leverages mesh representations to learn to accurately and efficiently simulate the dynamics of a wide range of physical systems.

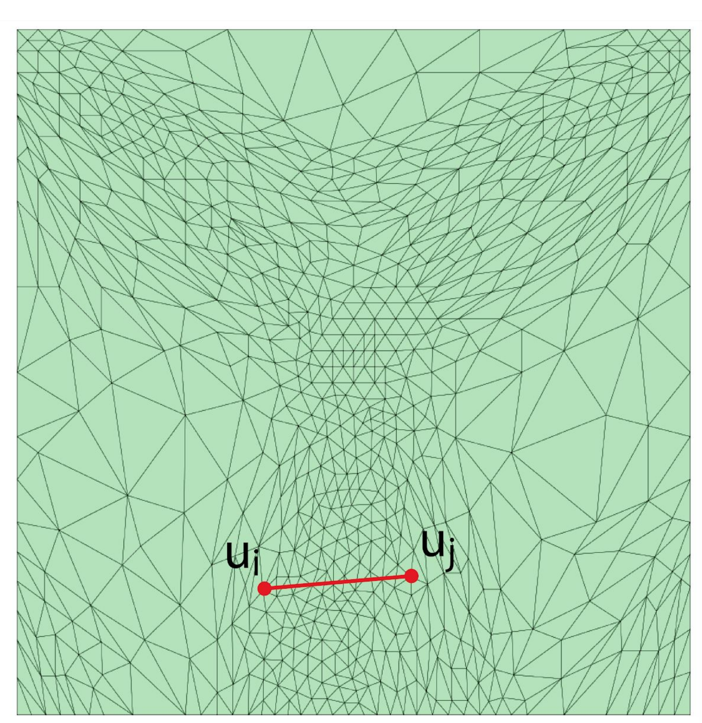
Model

We perform next-step prediction using a GraphNet with Encode-Process-Decode architecture. The **encoder** encodes the mesh at time t into a multigraph, the **processor** applies several steps of message passing on the graph, and the **decoder** decodes node attributes which can be integrated by the updater to obtain next-step predictions. We train on a single forward step, and at test time unroll the model for 1000s of steps.





World space



Mesh space

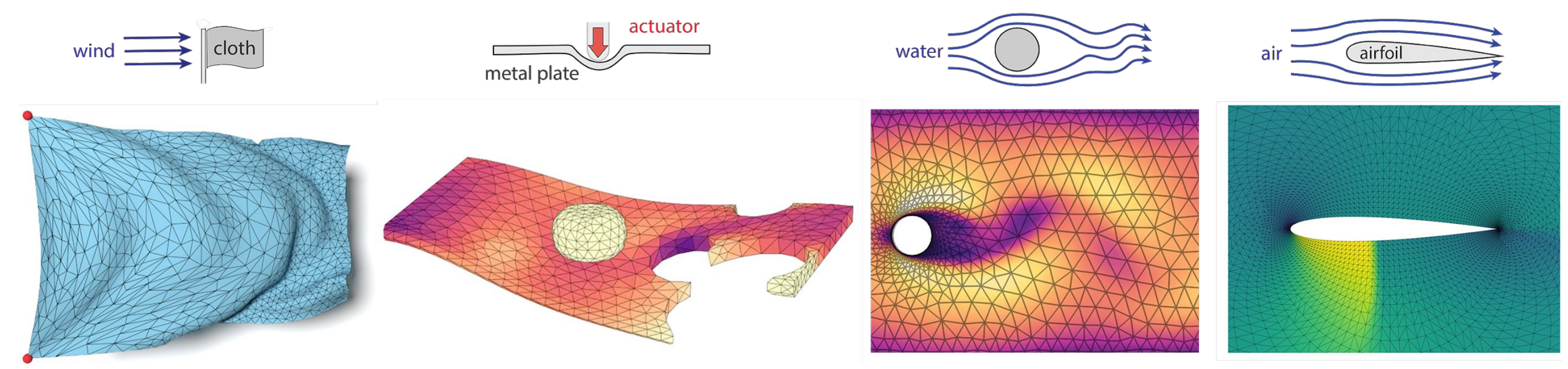
Notably, we compute and pass two types of messages:

Mesh edges follow the mesh connectivity and can efficiently evaluate internal dynamics, such as elasticity.

World edges are formed between nodes close in world-space but far in mesh space, and can be used to compute external dynamics such as collision and contact.

Experimental Domains

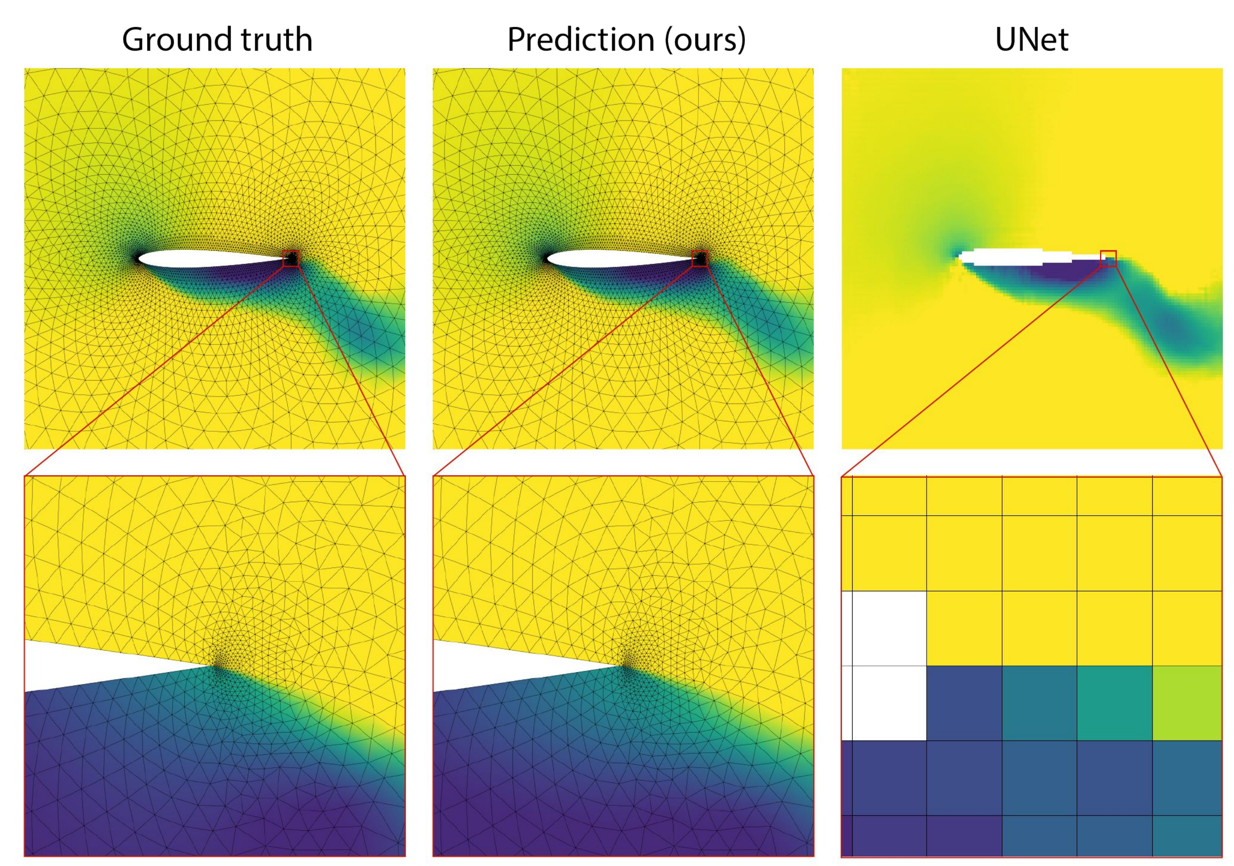
Our method is very flexible, and we show results on a wide range of physical systems: cloth, hyperelasticity, incompressible flow and compressible aerodynamics, generated by 3 different simulators. All systems are trained with the same model architecture.



Results

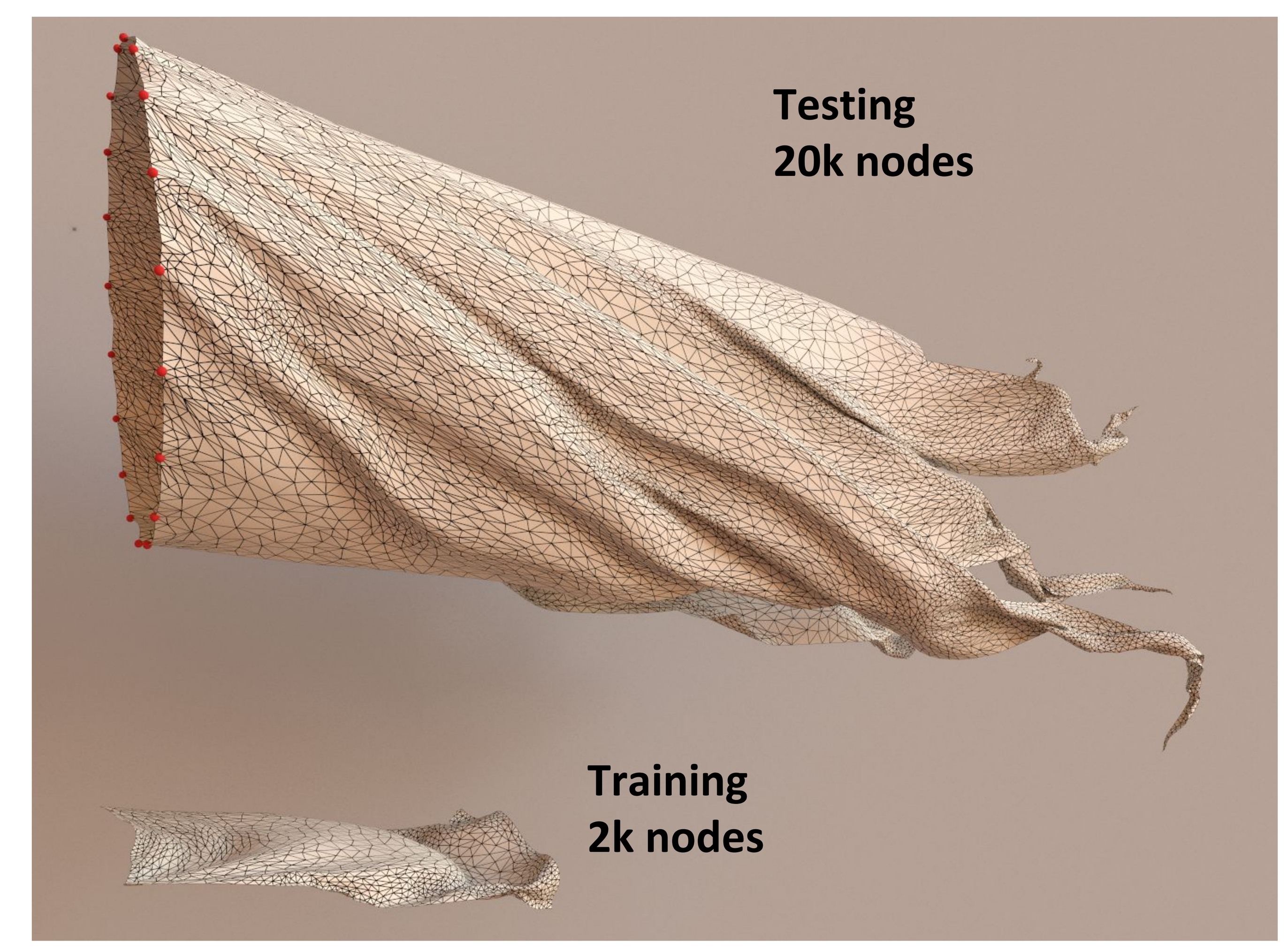
Meshes vs grids

We predict the flow around a x-section of an airfoil, and compare to a UNet baseline on a regular grid. By using irregular meshes, we are able to produce accurate predictions in the vital region around the wing tip. This region is under-resolved on the regular grid, even though is uses 8x more cells than our mesh.



Generalization

Our method generalized very well, even to systems far outside the training domain. Here, our model was trained on rectangular flags (bottom), and at test time applies to a windsock with tassels with 10x as many nodes.



Efficiency

Our method runs 10-290x faster than the ground truth simulators it was trained on, due to its ability to take larger timesteps and use of standard NN components which are optimized for hardware acceleration.

Full results

Paper: arxiv.org/abs/2010.03409

Videos: sites.google.com/view/meshgraphnets