Medical Imaging Intelligent Reality Lab

ASAN Medical Center

UNIVERSITY OF ULSAN

Learning to Simulate Complex Physics with Graph Networks

Alvaro Sanchez-Gonzalez [1], Jonathan Godwin[1], Tobias Pfaff [1], Rex Ying [1,2], Jure Leskovec [2], Peter W. Battaglia [1]

DeepMind | Department of Computer Science, Stanford University

Index

Introduction

UNIVERSITY OF ULSAN

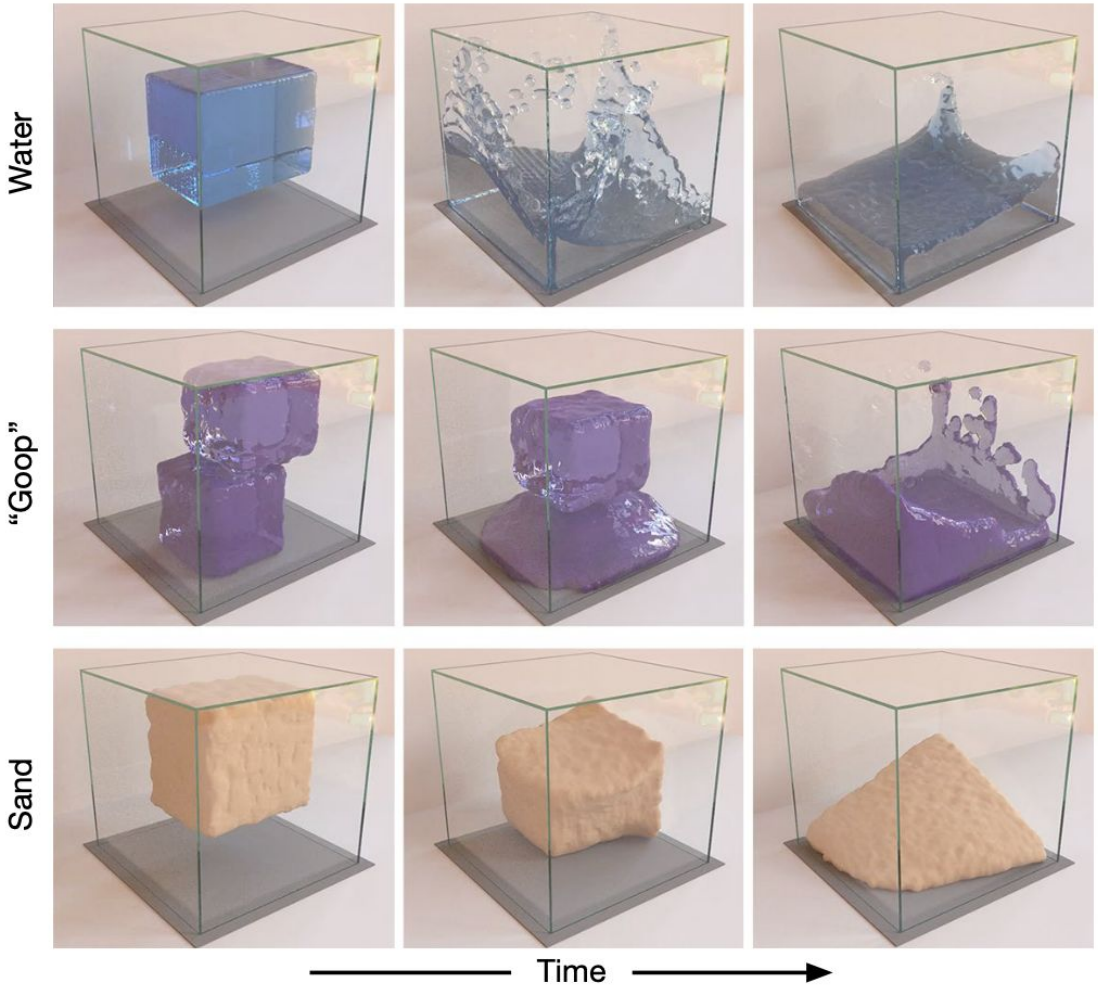
MI2RL Medical Imaging Intelligent Reality Lab

- Tradi- tional simulators can be **very expensive to create and use.** Building a simulator can entail years of engineering ef- fort, and **often must trade off generality for accuracy in a narrow range of settings.**

- **The large state spaces and complex dynamics have been difficult for standard end-to-end learning approaches to overcome.**

- While previous learning simulation approaches **have been highly specialized for particular tasks**,

→ We found our single GNS model performed well across dozens of experiments and was generally robust to hyperparameter choices.



Related Work

UNIVERSITY OF ULSAN

MIRL Medical Imaging Intelligent Reality Lab

Smoothed Particle Hydrodynamics (SPH)

- which evaluates pressure and vis cosity forces around each particle, and updates particles' velocities and positions accordingly

Position-based Dynamics (PBD)

- incompressibility and collision dynamics involve re solving pairwise distance constraints between particles, and directly predicting their position changes

Material Point Method (MPM)

- tiable particle-based simulators have recently appeared, e.g., DiffTaichi (Hu et al., 2019), PhiFlow (Holl et al., 2020), and Jax-MD (Schoenholz & Cubuk, 2019), which can backprop agate gradients through the architecture.

Related Work

Learning simulations from data

- Learning simulations from data (Grzeszczuk et al., 1998) has been an important area of study with applications in physics and graphics. Compared to engineered simulators, a learned simulator can be far more efficient for predicting complex phenomena (He et al., 2019); e.g., (Ladick`y et al., 2015; Wiewel et al., 2019) learn parts of a fluid simulator for faster prediction.

Graph Networks (GN)

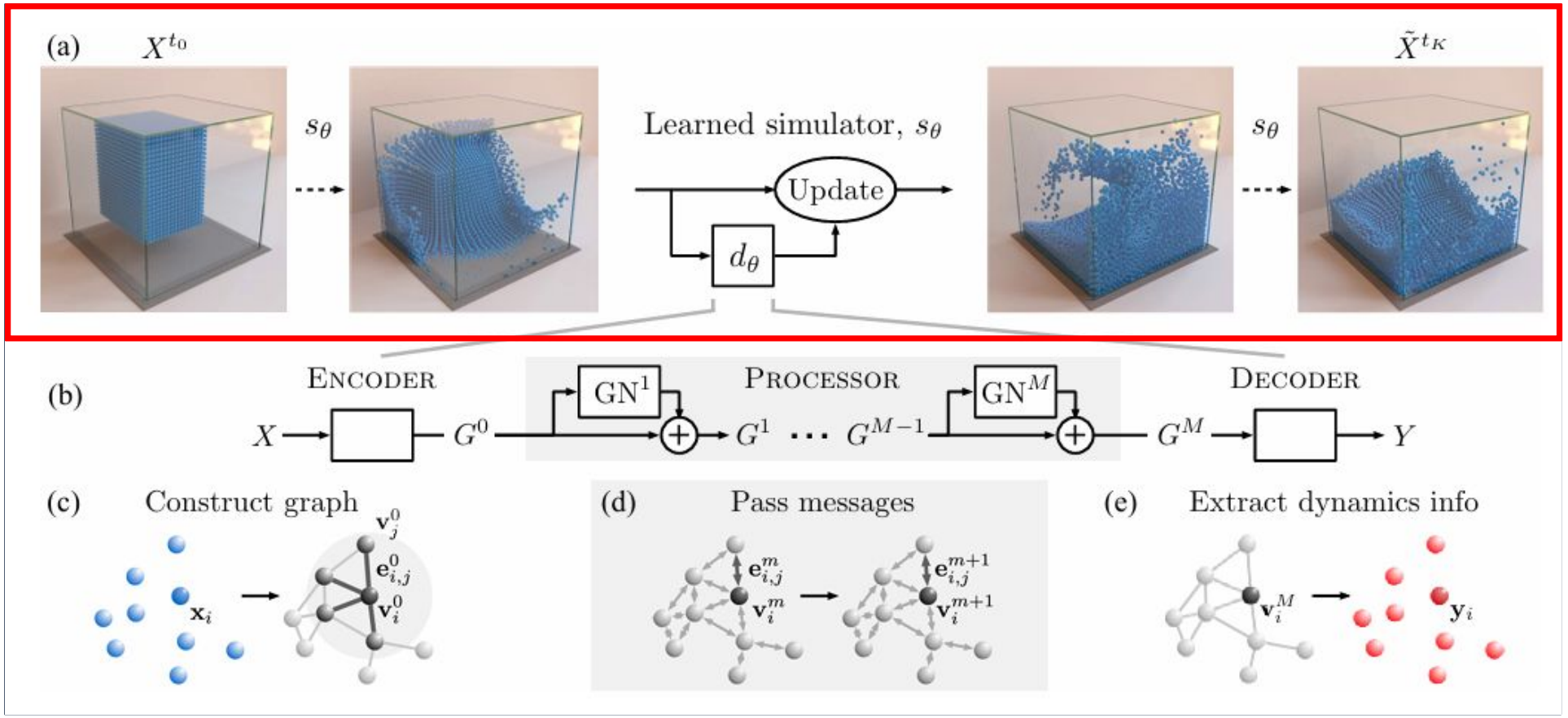
- Graph Neural Network have recently proven effective at learning forward dynamics in various set tings that involve interactions between many entities. Crucially, our GNS frame work is a general approach to learning simulation, is simpler to implement, and is more accurate across fluid, rigid, and deformable material systems.

GNS Model Framework

UNIVERSITY OF ULSAN

MIRL Medical Imaging Intelligent Reality Lab

3.1. General Learnable Simulation



$$X^t \in \mathcal{X}$$

$$\mathbf{X}^{t_{0:K}} = (X^{t_0}, \ldots, X^{t_K})$$

$$s : \mathcal{X} \to \mathcal{X}$$

$$d_\theta : \mathcal{X} \to \mathcal{Y}$$

$$\tilde{\mathbf{X}}^{t_{0:K}} = (X^{t_0}, \tilde{X}^{t_1}, \ldots, \tilde{X}^{t_K})$$

$$\tilde{X}^{t_{k+1}} = s(\tilde{X}^{t_k})$$

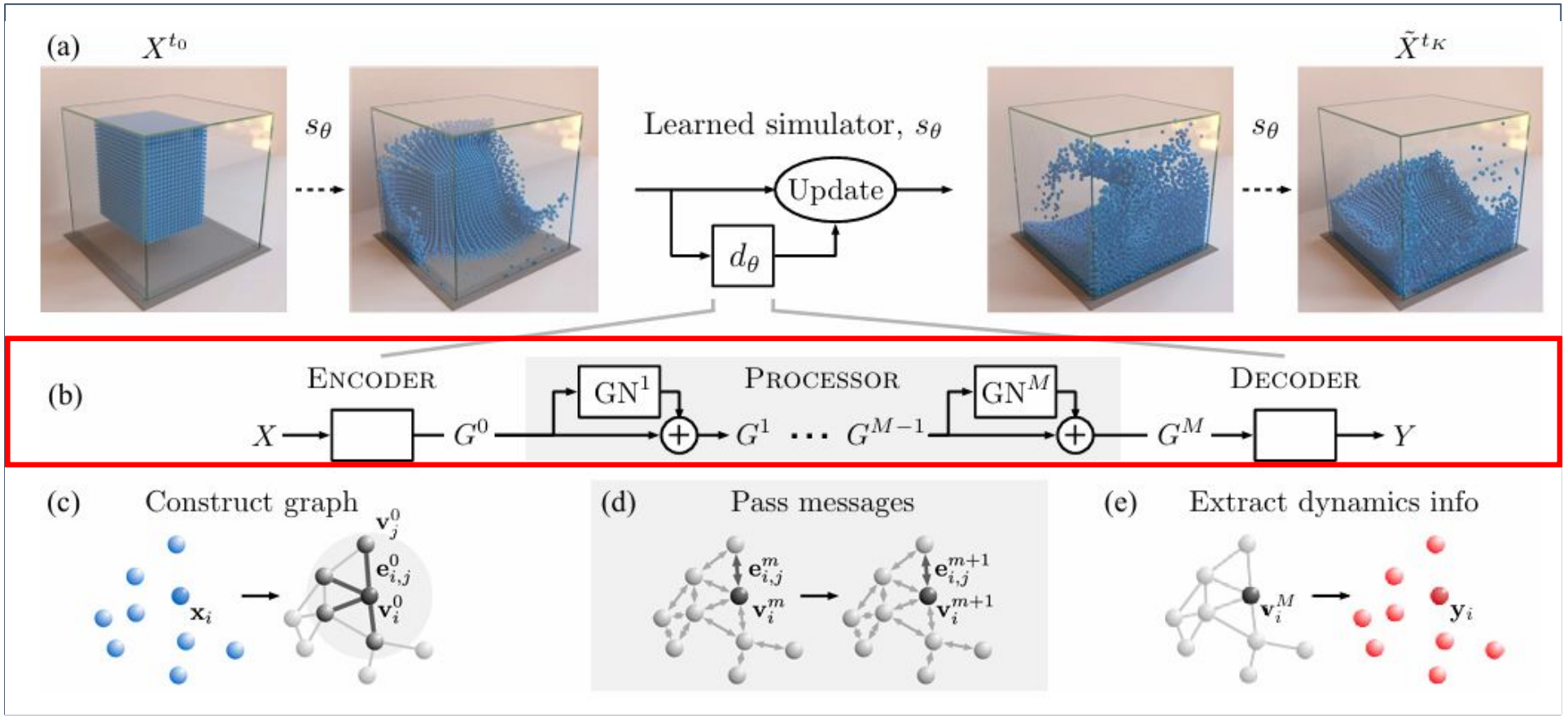
$$\tilde{X}^{t_{k+1}} = \text{Update}(\tilde{X}^{t_k}, d_\theta)$$

GNS Model Framework

UNIVERSITY OF ULSAN

MIRL Medical Imaging Intelligent Reality Lab

3.2. Simulation as Message-Passing on a Graph



The ENCODER : $\mathcal{X} \to \mathcal{G}$

The PROCESSOR : $\mathcal{G} \to \mathcal{G}$

The DECODER : $\mathcal{G} \to \mathcal{Y}$

4.1. Physical Domains

We explored how our GNS learns to simulate in datasets which contained three diverse, complex physical materials:

- water as a barely damped fluid, chaotic in nature;
- sand as a granular material with complex frictional behavior;
- and “goop” as a viscous, plastically deformable material.

UNIVERSITY OF ULSAN

MIRL Medical Imaging Intelligent Reality Lab

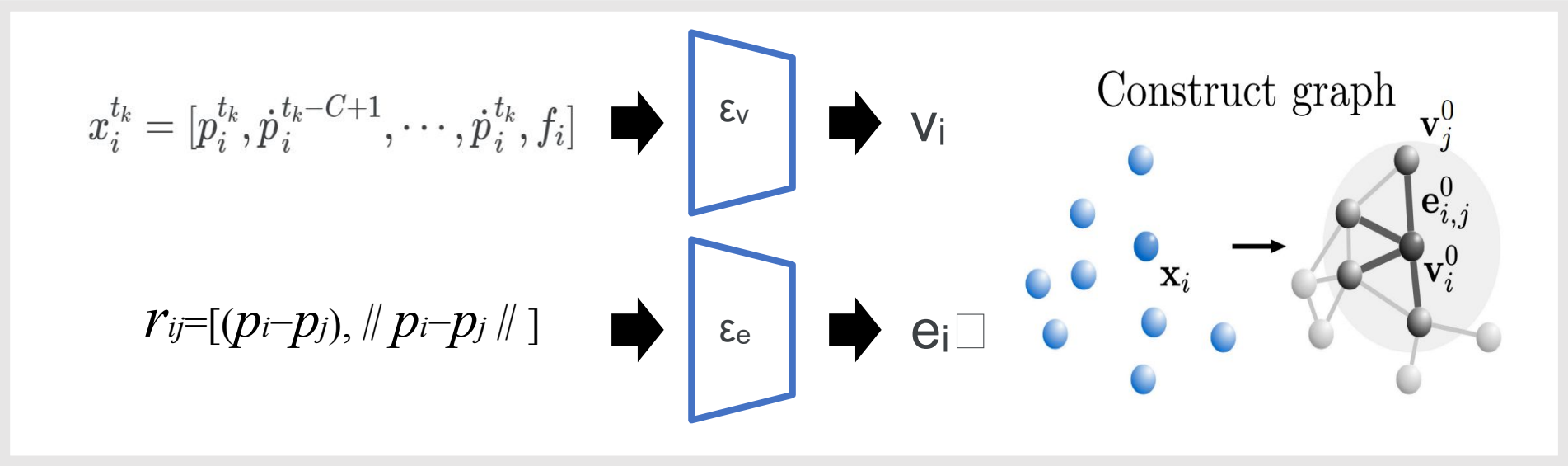
4.2. GNS Implementation Details

Input and output representations

Each particle's input state vector represents position, a sequence of C = 5 pre vious velocities1, and features that capture static material properties (e.g., water, sand, goop, rigid, boundary parti cle)

ENCODER details

The ENCODER constructs the graph structure G0 by assigning a node to each particle and adding edges between particles within a "connectivity radius", R, which reflected local interactions of particles, and which was kept constant for all simulations of the same resolution. The ENCODER implements v and e as multilayer percep trons (MLP), which encode node features and edge features into the latent vectors, vi and eij, of size 128.



UNIVERSITY OF ULSAN

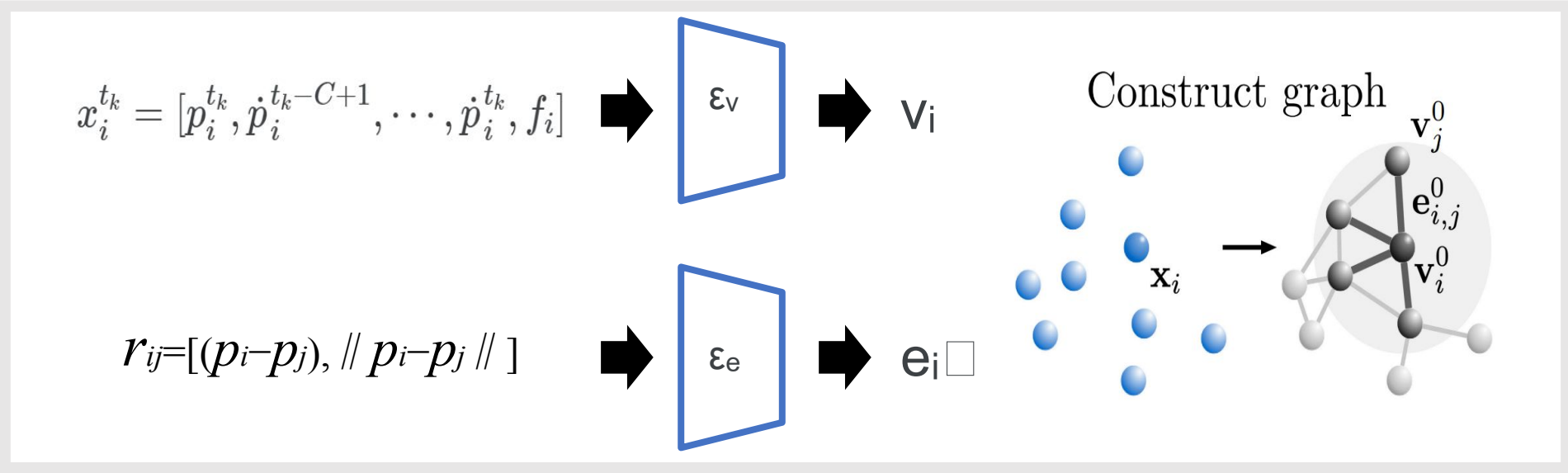
MIRL Medical Imaging Intelligent Reality Lab

4.2. GNS Implementation Details

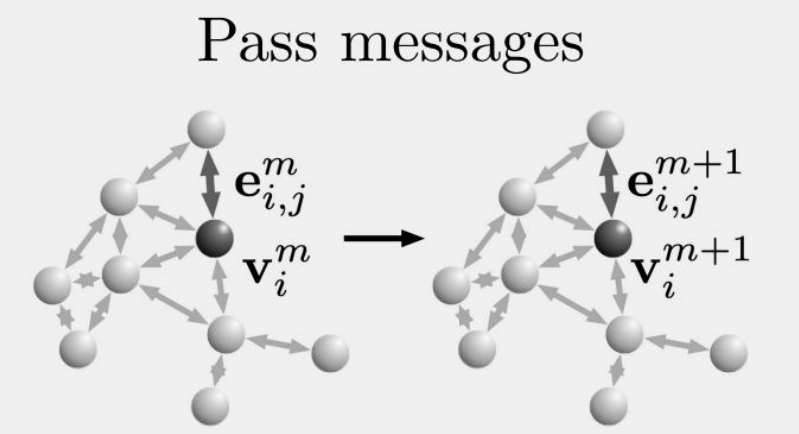
ENCODER details

Wetested two ENCODER variants, distinguished by whether they use absolute versus relative positional information.

The v was forced to ignore pi information within xi by masking it out. The e was provided with the relative positional displacement, and its magnitude2, $r_{ij}=[(p_i-p_j), \| p_i-p_j \|]$.

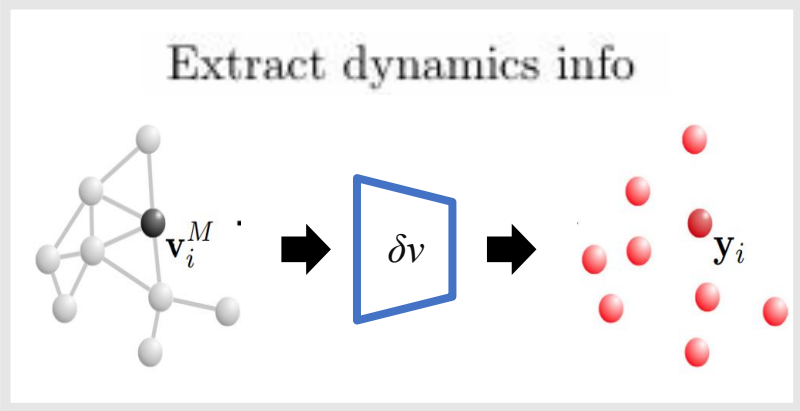


4.2. GNS Implementation Details



PROCESSOR details

We use GNs without global features or global updates (similar to an interaction network)3, and with a residual connections between the input and output latent node and edge attributes.



DECODER details.

Our decoder's learned function, v, is an MLP. After the DECODER, the future position and velocity are updated using an Euler integrator, so the yi corresponds to accelerations, pi, with 2D or 3D dimension, depending on the physical domain.

4.3. Training

Training noise

Modeling a complex and chaotic simula tion system requires the model to mitigate error accumu lation over long rollouts. Because we train our models on ground-truth one-step data, they are never presented with input data corrupted by this sort of accumulated noise.

Normalization

We normalize all input and target vectors elementwise to zero mean and unit variance, using statistics computed online during training.

Loss function and optimization procedures.

We ran domly sampled particle state pairs $(x^t k_i, x^t k+1_i)$ from train ing trajectories, calculated target accelerations $\ddot{p}^t k_i$, *and computed the L2 loss on the predicted per-particle accel erations, i.e.,*

$$L(x^t k_i, x^t k+1_i; \theta) = \| d_\theta(x^t k_i) - \ddot{p}^t k_i \|_2$$

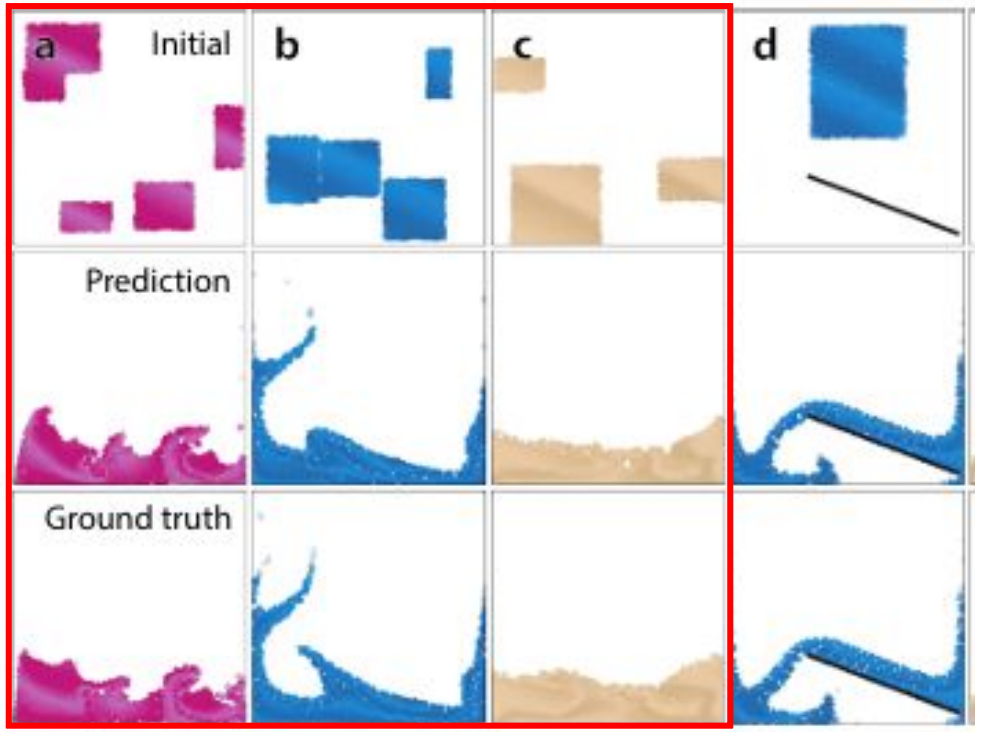
4.4. Evaluation

To report quantitative results, we evaluated our models af ter training converged by computing one-step and rollout metrics on held-out test trajectories, drawn from the same distribution of initial conditions used for training. We used particle-wise MSE as our main metric between ground truth and predicted data, both for rollout and one-step predictions, averaging across time, particle and spatial axes.

5.1. Simulating Complex Materials

Our GNSmodel was very effective at learning to simulate different complex materials. Table 1 shows the one-step and rollout accuracy, as MSE, for all experiments. For intuition about what these numbers mean, the edge length of the container was approximately 10, and Figure 3(a-c) shows rendered images of the rollouts of our model, compared to ground truth6.

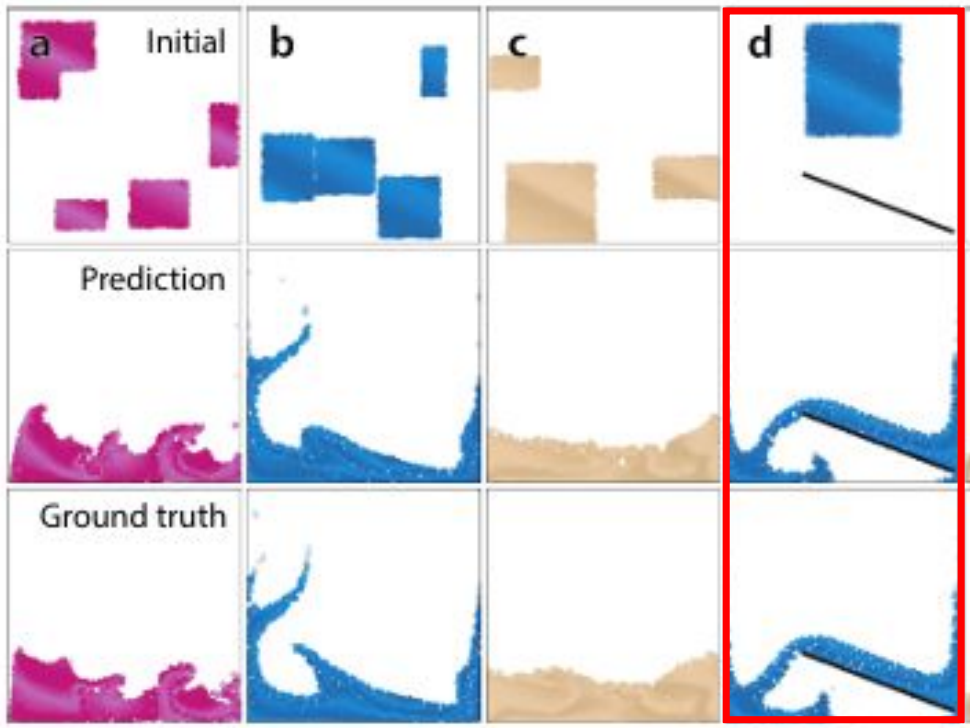
Experimental domain	N	K	**1-step** ($\times 10^{-9}$)	**Rollout** ($\times 10^{-3}$)
WATER-3D (SPH)	13k	800	8.66	10.1
SAND-3D	20k	350	1.42	0.554
GOOP-3D	14k	300	1.32	0.618
WATER-3D-S (SPH)	5.8k	800	9.66	9.52
BOXBATH (PBD)	1k	150	54.5	4.2
WATER	1.9k	1000	2.82	17.4
SAND	2k	320	6.23	2.37
GOOP	1.9k	400	2.91	1.89
MULTIMATERIAL	2k	1000	1.81	16.9
FLUIDSHAKE	1.3k	2000	2.1	20.1
WATERDROP	1k	1000	1.52	7.01
WATERDROP-XL	7.1k	1000	1.23	14.9
WATERRAMPS	2.3k	600	4.91	11.6
SANDRAMPS	3.3k	400	2.77	2.07
RANDOMFLOOR	3.4k	600	2.77	6.72
CONTINUOUS	4.3k	400	2.06	1.06



5.1. Simulating Complex Materials

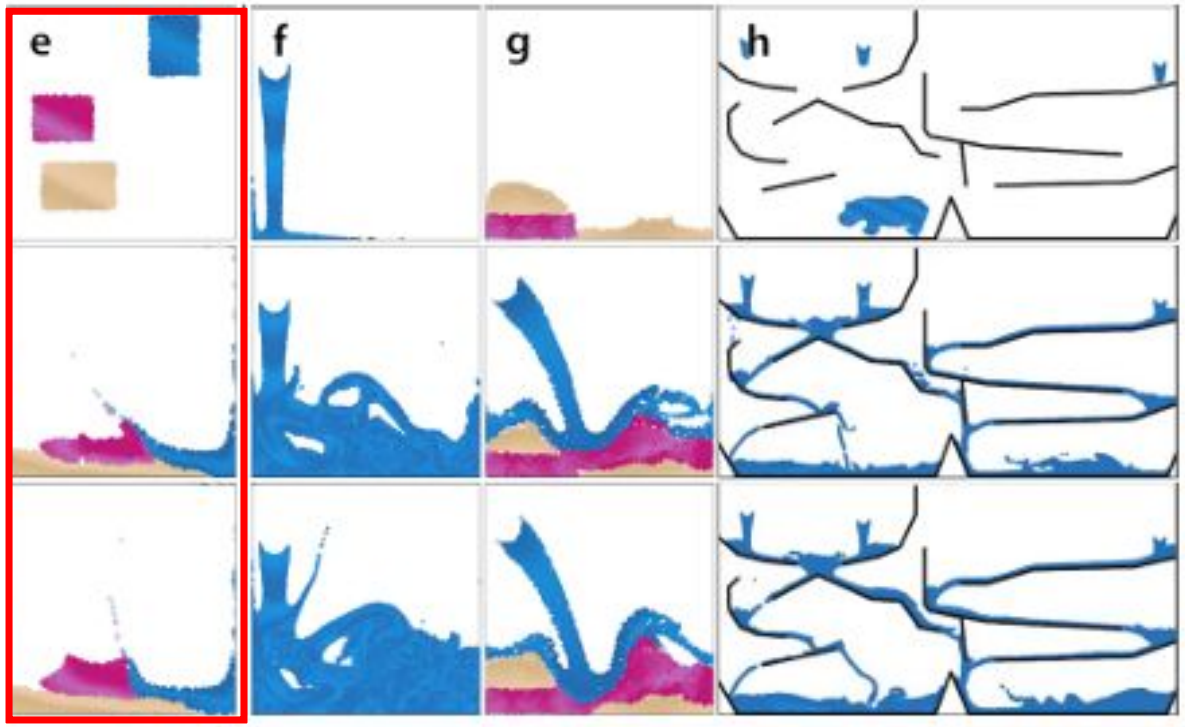
Our GNSmodel was very effective at learning to simulate different complex materials. Table 1 shows the one-step and rollout accuracy, as MSE, for all experiments. For intuition about what these numbers mean, the edge length of the container was approximately 10, and Figure 3(a-c) shows rendered images of the rollouts of our model, compared to ground truth6.

Experimental domain	N	K	**1-step** ($\times 10^{-9}$)	**Rollout** ($\times 10^{-3}$)
WATER-3D (SPH)	13k	800	8.66	10.1
SAND-3D	20k	350	1.42	0.554
GOOP-3D	14k	300	1.32	0.618
WATER-3D-S (SPH)	5.8k	800	9.66	9.52
BOXBATH (PBD)	1k	150	54.5	4.2
WATER	1.9k	1000	2.82	17.4
SAND	2k	320	6.23	2.37
GOOP	1.9k	400	2.91	1.89
MULTIMATERIAL	2k	1000	1.81	16.9
FLUIDSHAKE	1.3k	2000	2.1	20.1
WATERDROP	1k	1000	1.52	7.01
WATERDROP-XL	7.1k	1000	1.23	14.9
WATERRAMPS	2.3k	600	4.91	11.6
SANDRAMPS	3.3k	400	2.77	2.07
RANDOMFLOOR	3.4k	600	2.77	6.72
CONTINUOUS	4.3k	400	2.06	1.06



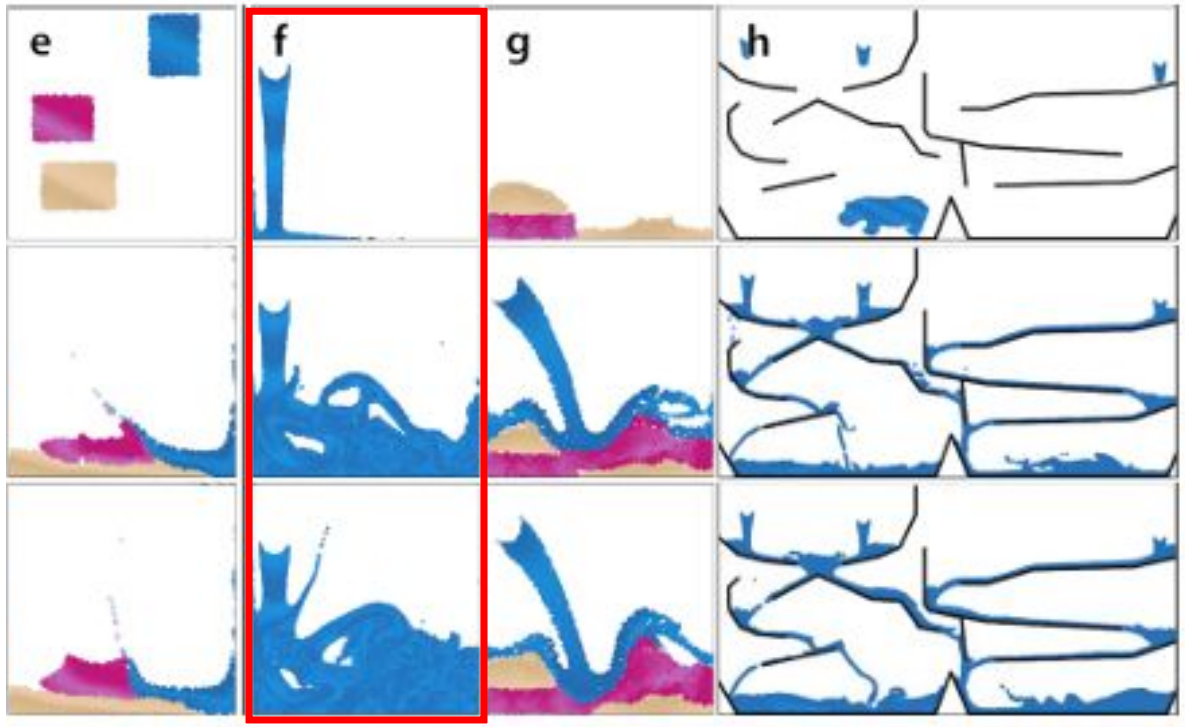
5.2. Multiple Interacting Materials & Generalization

Experimental domain	N	K	1-step ($\times 10^{-9}$)	Rollout ($\times 10^{-3}$)
WATER-3D (SPH)	13k	800	8.66	10.1
SAND-3D	20k	350	1.42	0.554
GOOP-3D	14k	300	1.32	0.618
WATER-3D-S (SPH)	5.8k	800	9.66	9.52
BOXBATH (PBD)	1k	150	54.5	4.2
WATER	1.9k	1000	2.82	17.4
SAND	2k	320	6.23	2.37
GOOP	1.9k	400	2.91	1.89
MULTIMATERIAL	2k	1000	1.81	16.9
FLUIDSHAKE	1.3k	2000	2.1	20.1
WATERDROP	1k	1000	1.52	7.01
WATERDROP-XL	7.1k	1000	1.23	14.9
WATERRAMPS	2.3k	600	4.91	11.6
SANDRAMPS	3.3k	400	2.77	2.07
RANDOMFLOOR	3.4k	600	2.77	6.72
CONTINUOUS	4.3k	400	2.06	1.06



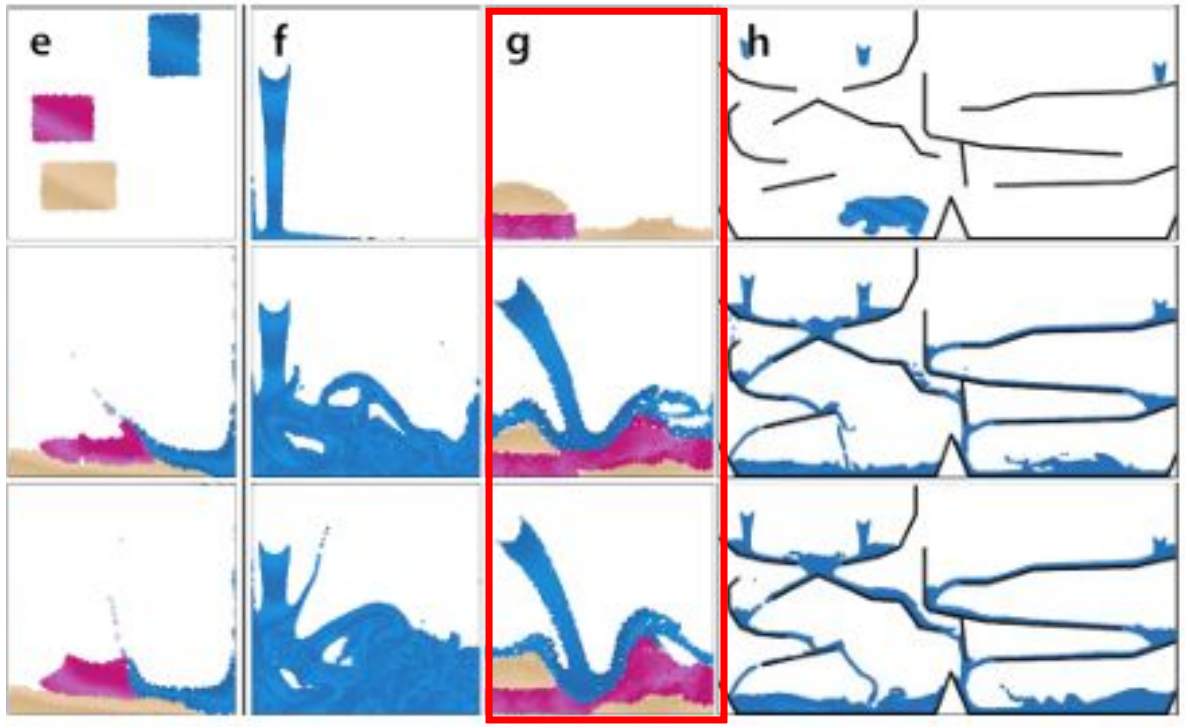
5.2. Multiple Interacting Materials & Generalization

Experimental domain	N	K	1-step ($\times 10^{-9}$)	Rollout ($\times 10^{-3}$)
WATER-3D (SPH)	13k	800	8.66	10.1
SAND-3D	20k	350	1.42	0.554
GOOP-3D	14k	300	1.32	0.618
WATER-3D-S (SPH)	5.8k	800	9.66	9.52
BOXBATH (PBD)	1k	150	54.5	4.2
WATER	1.9k	1000	2.82	17.4
SAND	2k	320	6.23	2.37
GOOP	1.9k	400	2.91	1.89
MULTIMATERIAL	2k	1000	1.81	16.9
FLUIDSHAKE	1.3k	2000	2.1	20.1
WATERDROP	1k	1000	1.52	7.01
WATERDROP-XL	7.1k	1000	1.23	14.9
WATERRAMPS	2.3k	600	4.91	11.6
SANDRAMPS	3.3k	400	2.77	2.07
RANDOMFLOOR	3.4k	600	2.77	6.72
CONTINUOUS	4.3k	400	2.06	1.06



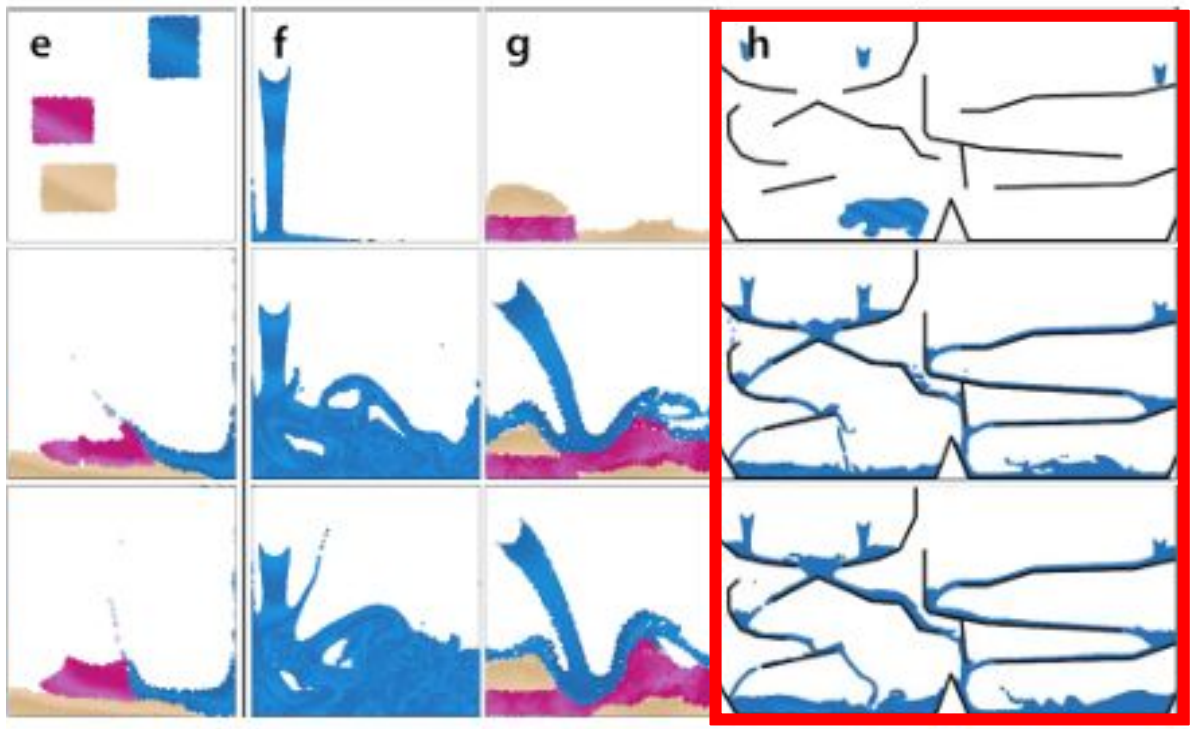
5.2. Multiple Interacting Materials & Generalization

Experimental domain	N	K	1-step ($\times 10^{-9}$)	Rollout ($\times 10^{-3}$)
WATER-3D (SPH)	13k	800	8.66	10.1
SAND-3D	20k	350	1.42	0.554
GOOP-3D	14k	300	1.32	0.618
WATER-3D-S (SPH)	5.8k	800	9.66	9.52
BOXBATH (PBD)	1k	150	54.5	4.2
WATER	1.9k	1000	2.82	17.4
SAND	2k	320	6.23	2.37
GOOP	1.9k	400	2.91	1.89
MULTIMATERIAL	2k	1000	1.81	16.9
FLUIDSHAKE	1.3k	2000	2.1	20.1
WATERDROP	1k	1000	1.52	7.01
WATERDROP-XL	7.1k	1000	1.23	14.9
WATERRAMPS	2.3k	600	4.91	11.6
SANDRAMPS	3.3k	400	2.77	2.07
RANDOMFLOOR	3.4k	600	2.77	6.72
CONTINUOUS	4.3k	400	2.06	1.06



5.2. Multiple Interacting Materials & Generalization

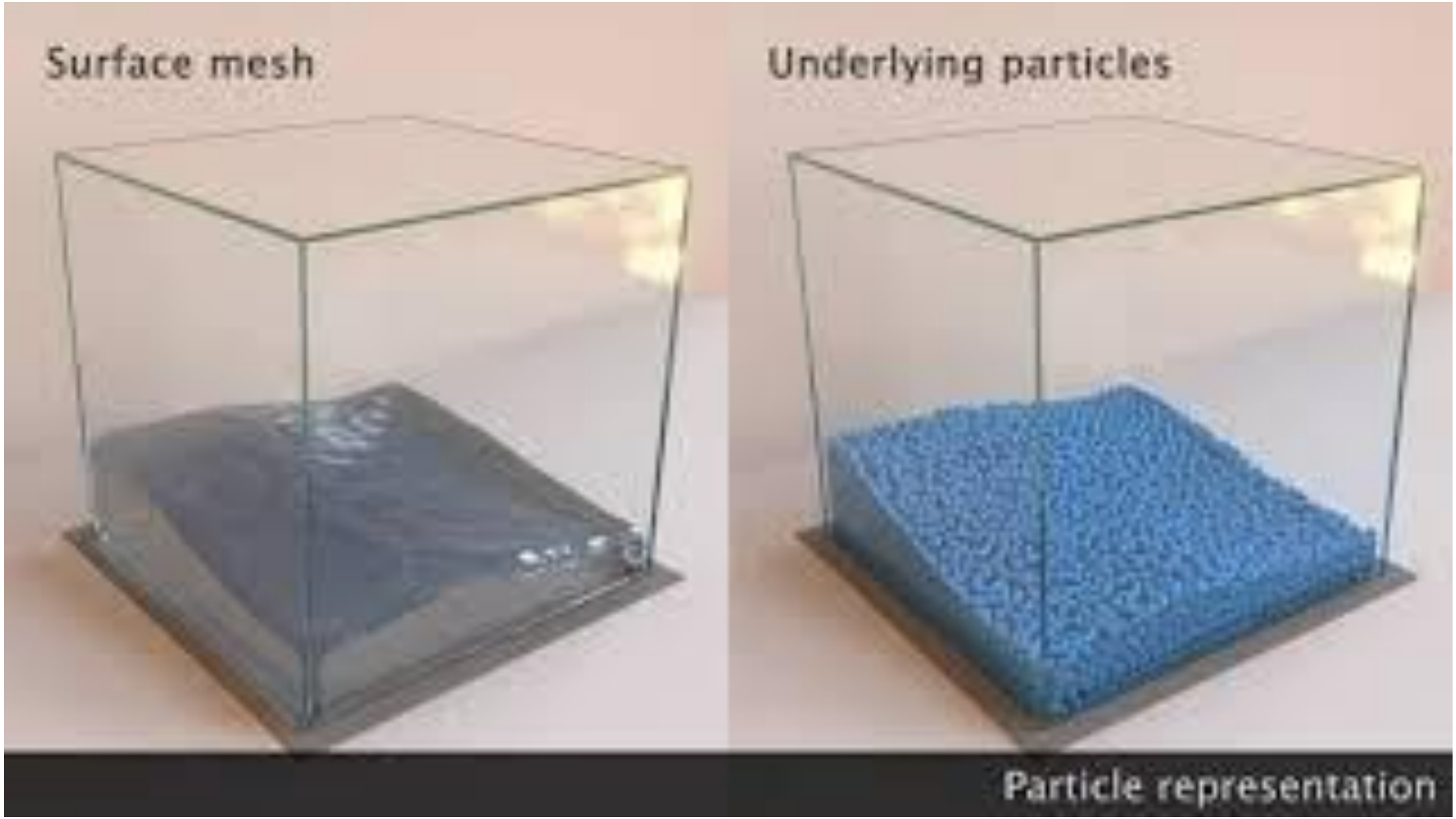
Experimental domain	N	K	1-step ($\times 10^{-9}$)	Rollout ($\times 10^{-3}$)
WATER-3D (SPH)	13k	800	8.66	10.1
SAND-3D	20k	350	1.42	0.554
GOOP-3D	14k	300	1.32	0.618
WATER-3D-S (SPH)	5.8k	800	9.66	9.52
BOXBATH (PBD)	1k	150	54.5	4.2
WATER	1.9k	1000	2.82	17.4
SAND	2k	320	6.23	2.37
GOOP	1.9k	400	2.91	1.89
MULTIMATERIAL	2k	1000	1.81	16.9
FLUIDSHAKE	1.3k	2000	2.1	20.1
WATERDROP	1k	1000	1.52	7.01
WATERDROP-XL	7.1k	1000	1.23	14.9
WATERRAMPS	2.3k	600	4.91	11.6
SANDRAMPS	3.3k	400	2.77	2.07
RANDOMFLOOR	3.4k	600	2.77	6.72
CONTINUOUS	4.3k	400	2.06	1.06



Result

UNIVERSITY OF ULSAN

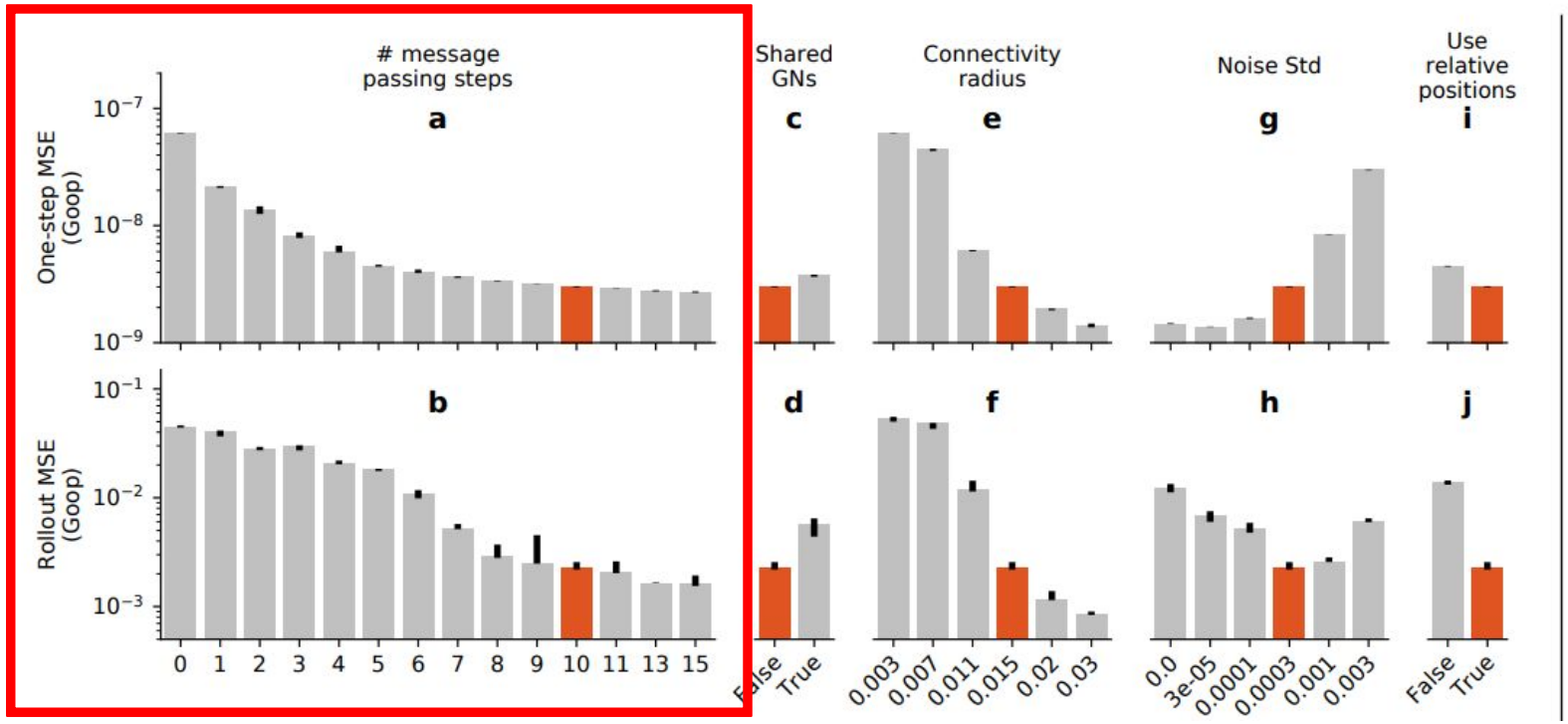
MI2RL Medical Imaging Intelligent Reality Lab



UNIVERSITY OF ULSAN

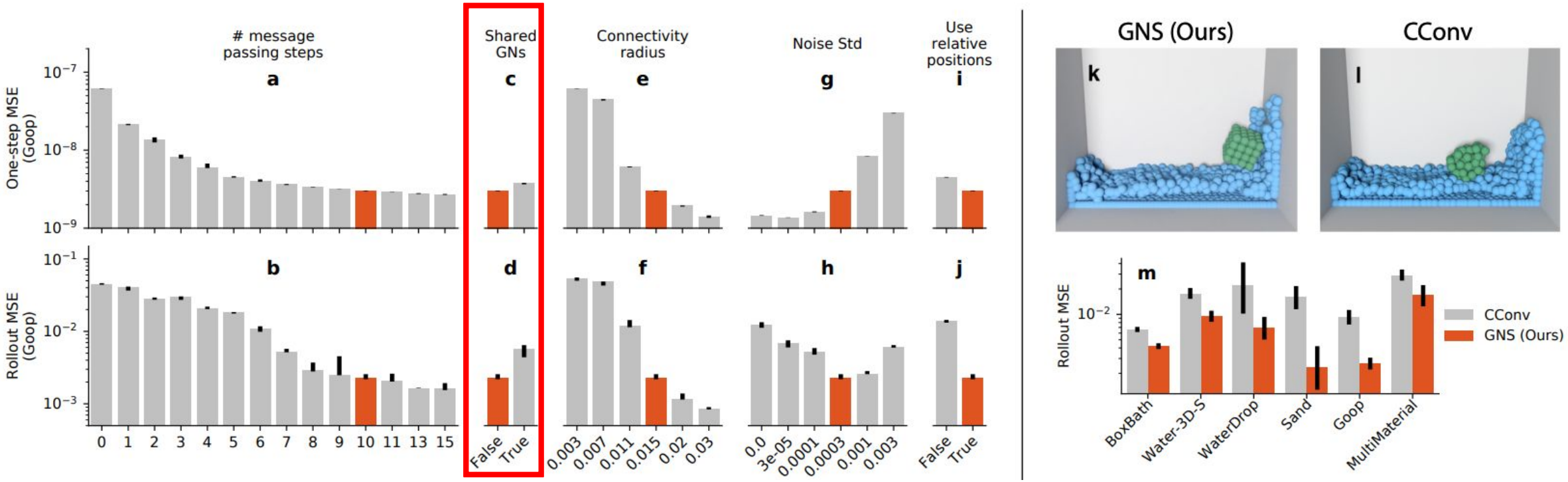
MIRL Medical Imaging Intelligent Reality Lab

5.4. Key Architectural Choices



the number of message-passing steps

5.4. Key Architectural Choices

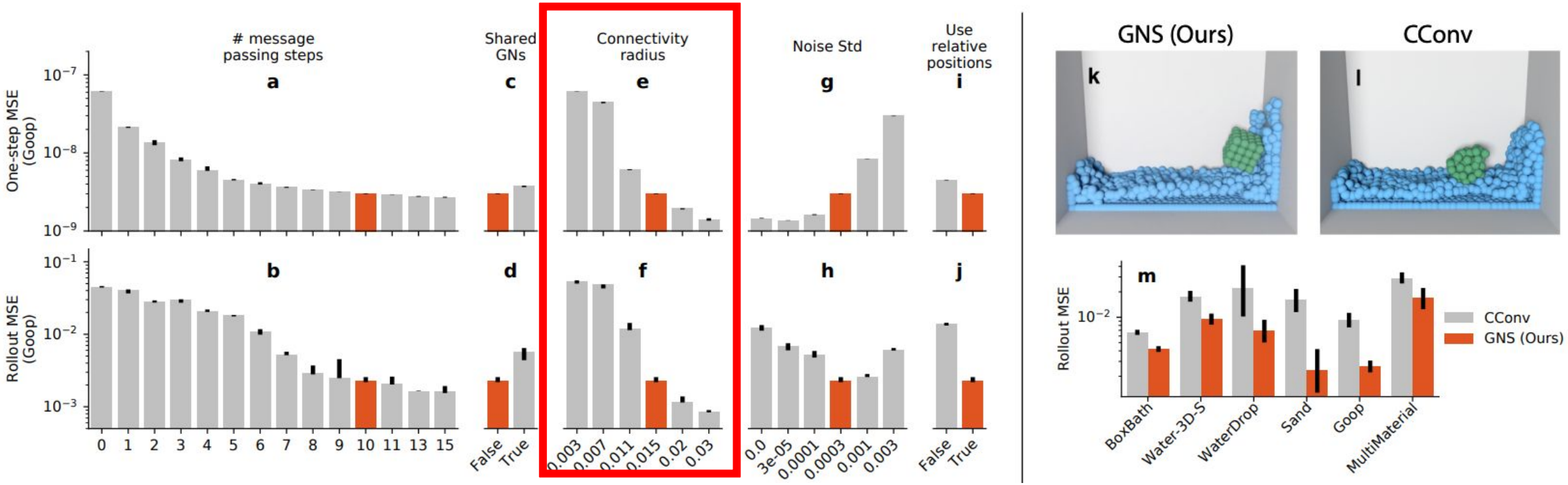


shared vs. **unshared PROCESSOR GN parameters**

UNIVERSITY OF ULSAN

MIRL Medical Imaging Intelligent Reality Lab

5.4. Key Architectural Choices

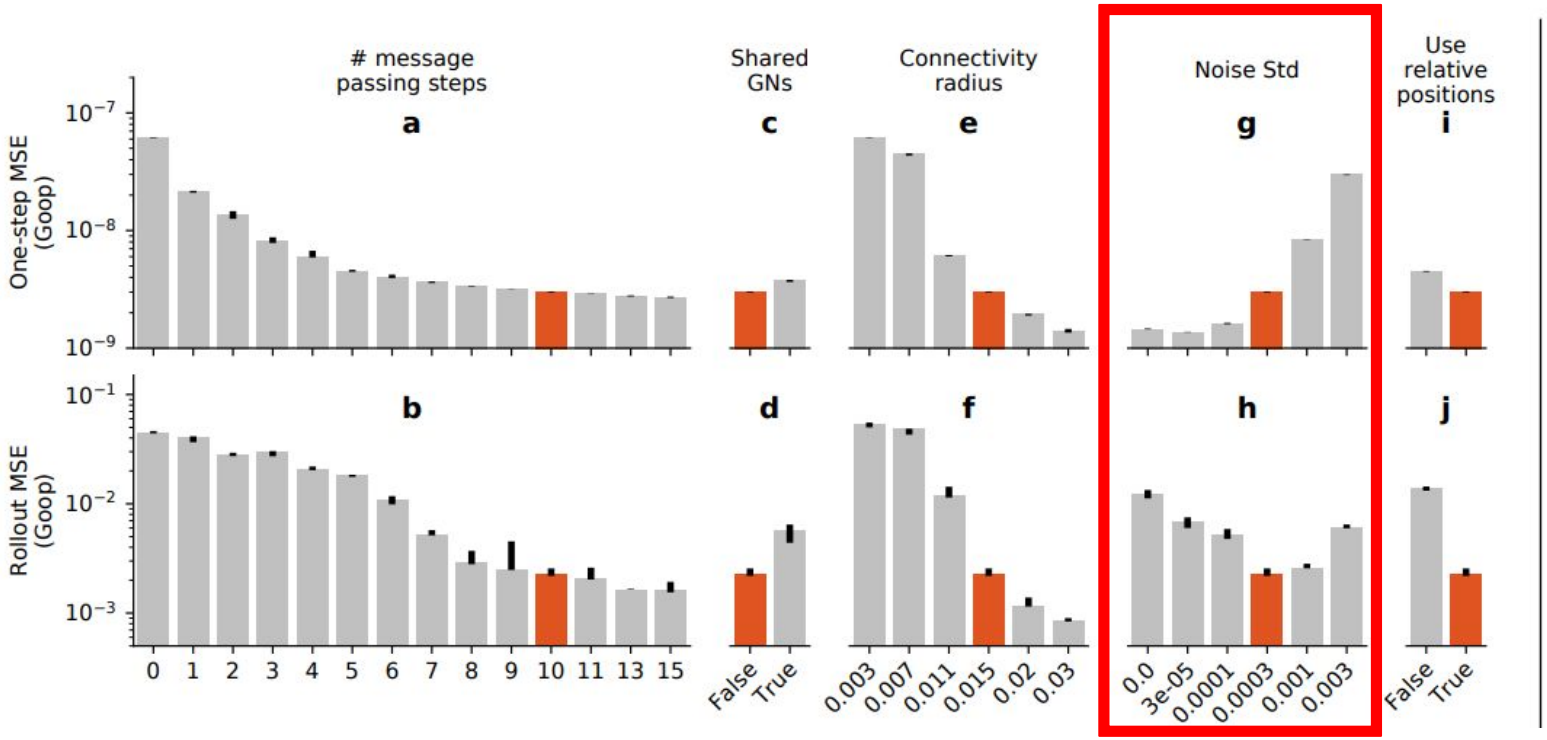


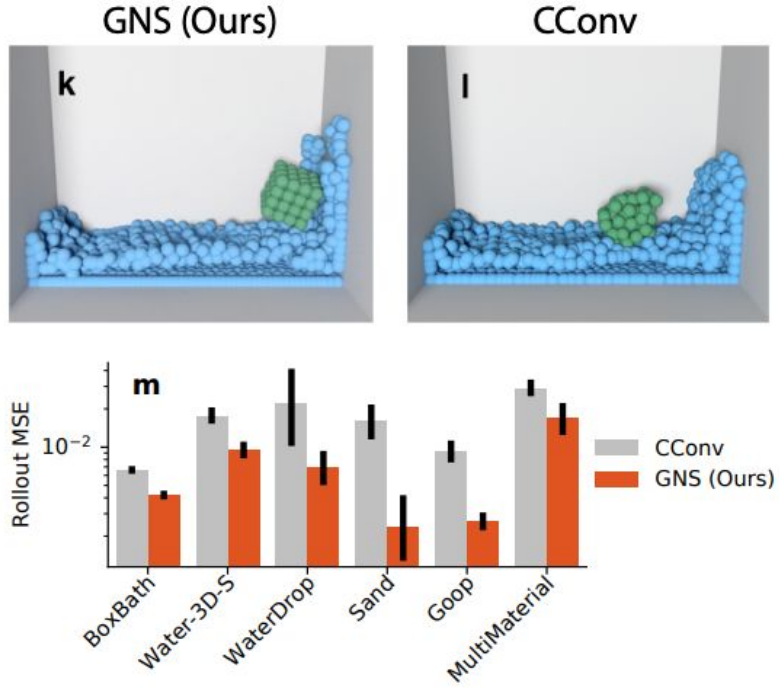
the connectivity radius

UNIVERSITY OF ULSAN

MIRL Medical Imaging Intelligent Reality Lab

5.4. Key Architectural Choices



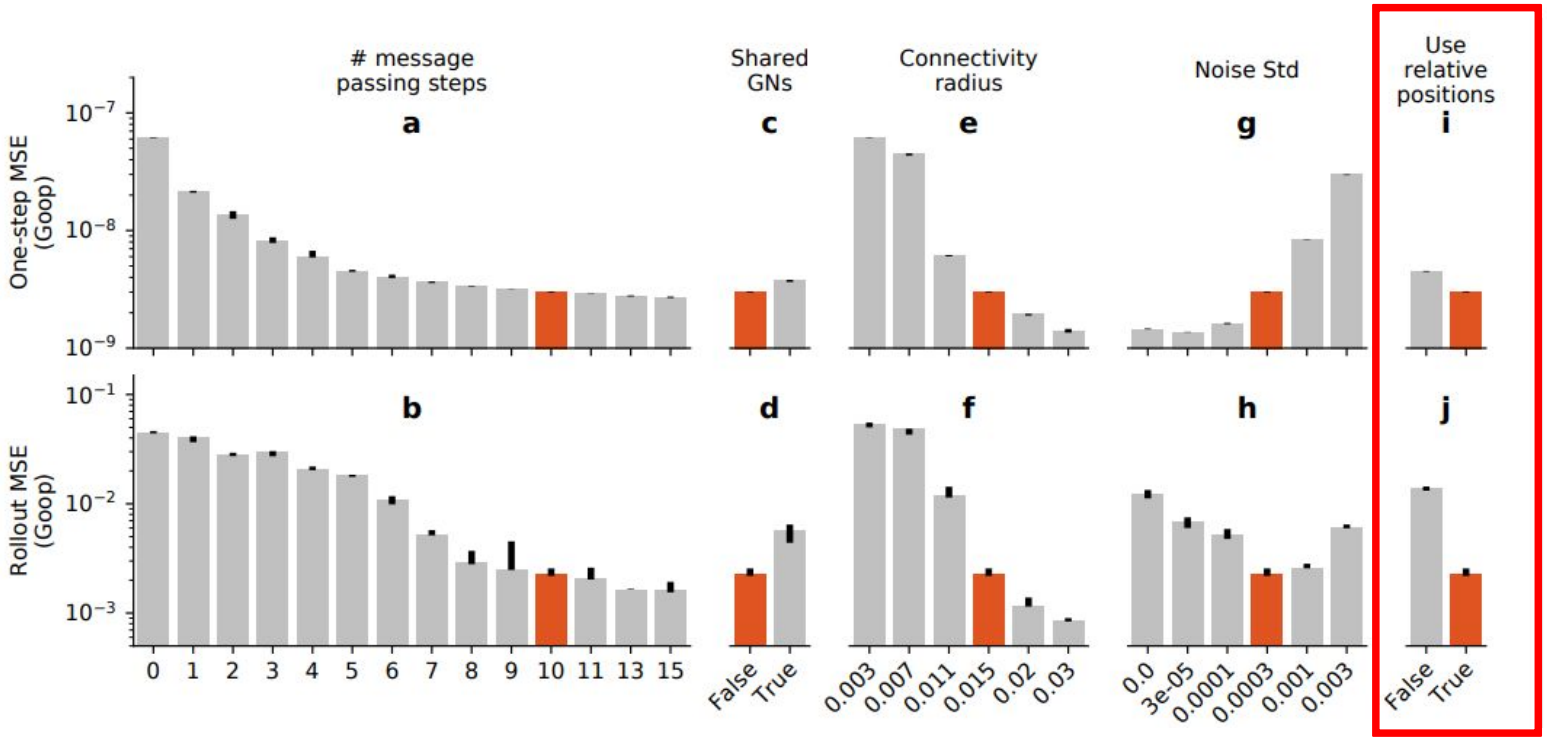


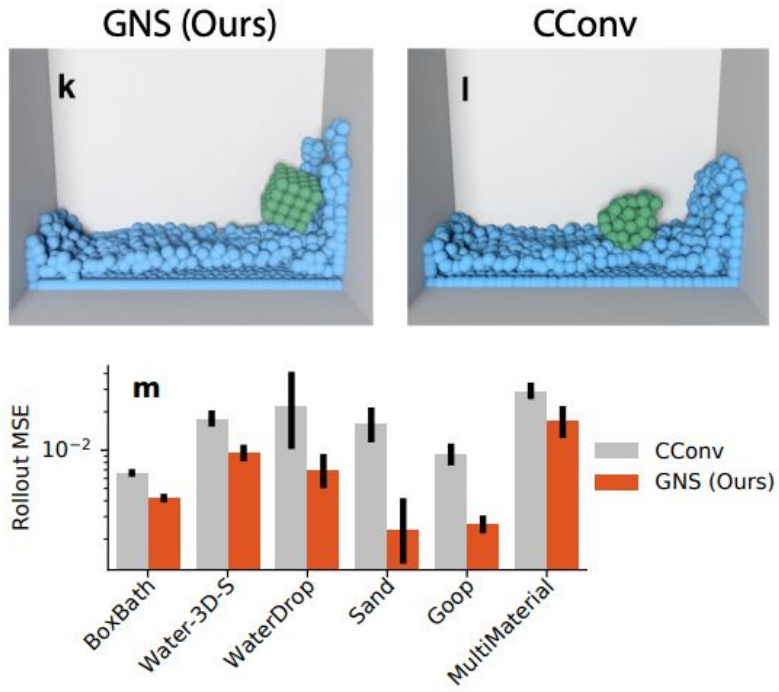
the scale of noise added to the inputs during training

UNIVERSITY OF ULSAN

MIRL Medical Imaging Intelligent Reality Lab

5.4. Key Architectural Choices



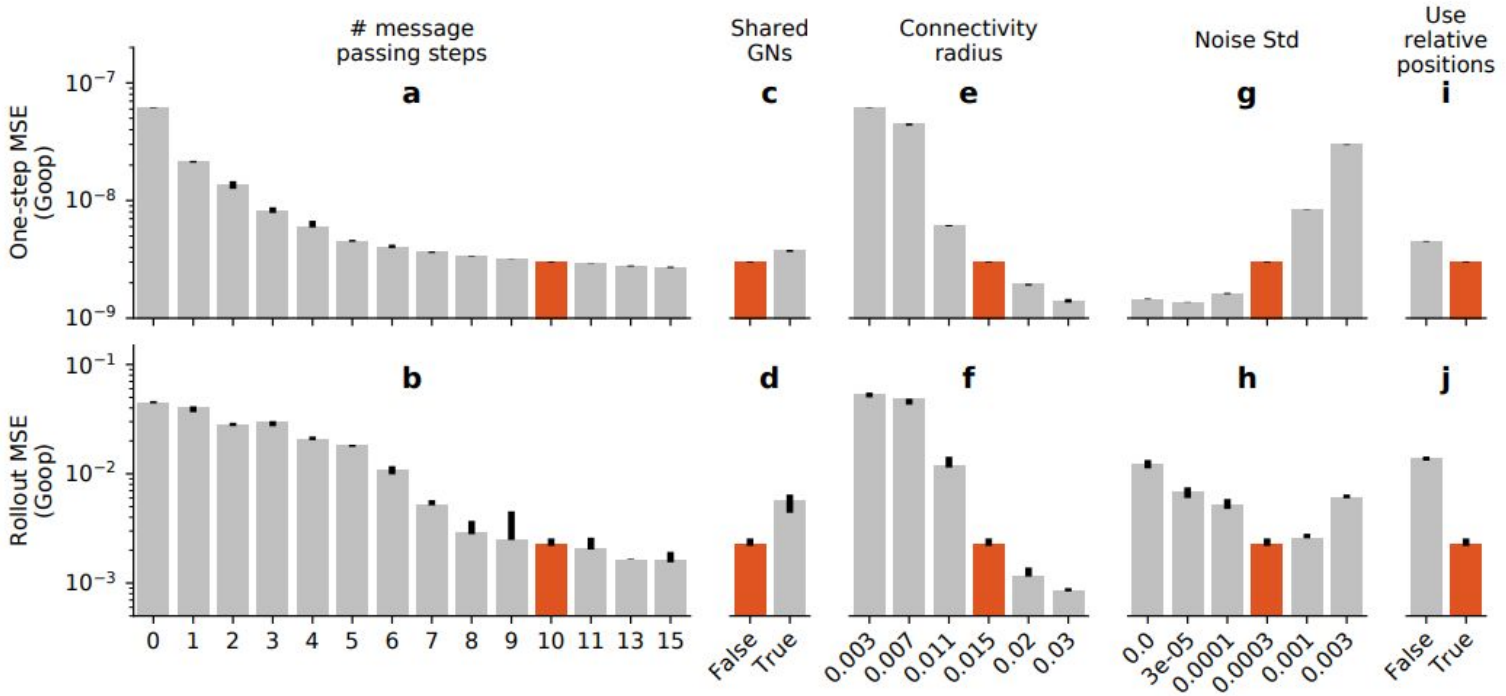


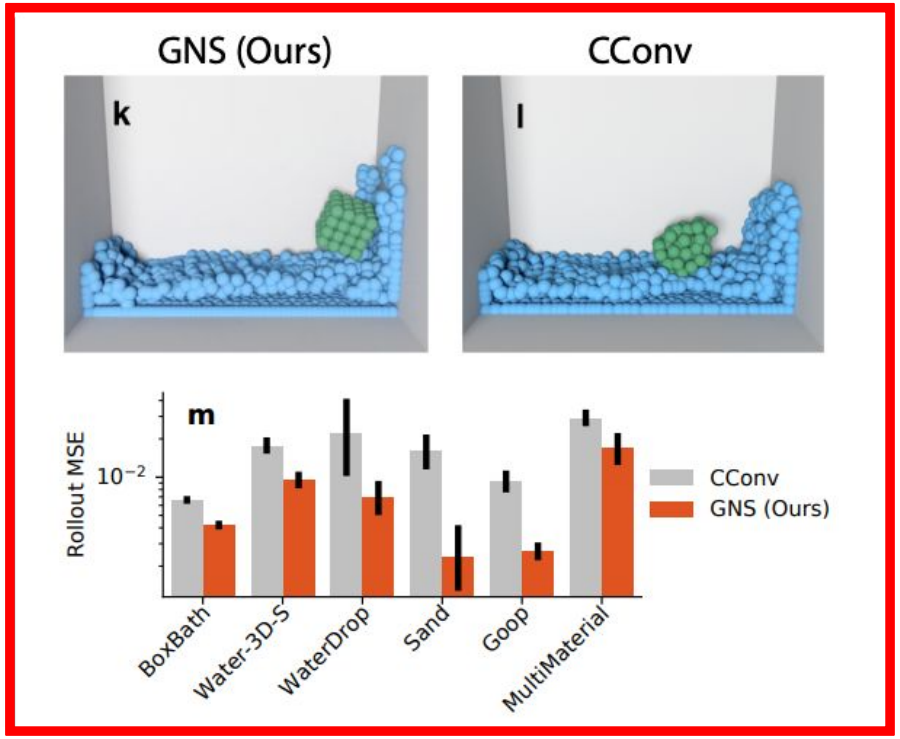
relative vs. **absolute ENCODER**

UNIVERSITY OF ULSAN

MI2RL Medical Imaging Intelligent Reality Lab

5.4. Key Architectural Choices





CConv vs. **GNS**

Conclusion

UNIVERSITY OF ULSAN

MI2RL Medical Imaging Intelligent Reality Lab

- We presented a powerful machine learning frame work for learning to simulate complex systems, based on particle based representations of physics and learned message passing on graphs.
- Our experimental results show our single GNS architecture can learn to simulate the dynamics of fluids, rigid solids, and deformable materials, interacting with on eanother, using tens of thousands of particles over thousands time steps.
- More broadly, this work is a key advance toward more sophisticated generative models, and furnishes the modern AI toolkit with a greater capacity for physical reasoning.

UNIVERSITY OF ULSAN

ASAN FOUNDATION
아산재단

ASAN
Medical Center

Thank you

MI2RL
Medical
Imaging
Intelligent
Reality
Lab