Learning Controllable Adaptive Simulation for Multi-scale Physics

Tailin Wu[1]*, Takashi Maruyama[1,2]*, Qingqing Zhao[1]*, Gordon Wetzstein[1], Jure Leskovec[1]

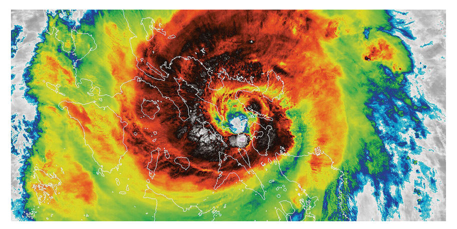
[1]Stanford University, [2]NEC Corporation

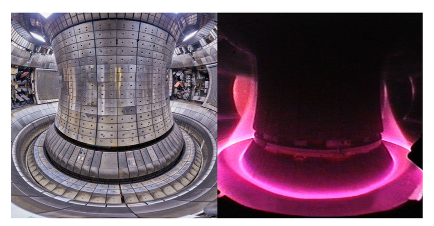
AI4Science Workshop

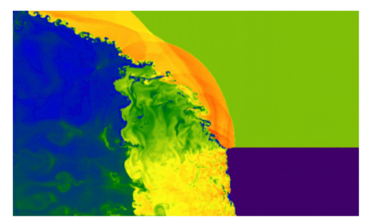
Motivation

Many physical systems in science and engineering are multi-resolution:

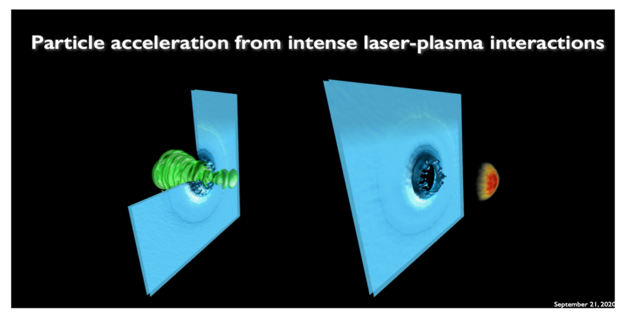
- A small fraction of the system is highly dynamic, and requires very fine-grained resolution to simulate accurately
- While a majority of the system is changing slowly


Weather forecasting


Disruptive instabilities in nuclear fusion


Modeling Compressible Reactive Gas Dynamics

E.g. In laser-plasma interaction, only ~0.01% of the particles are accelerated but can carry 10-50% of system energy.



The goal is to simulate the evolution of such multi-resolution systems accurately and efficiently.

Prior works

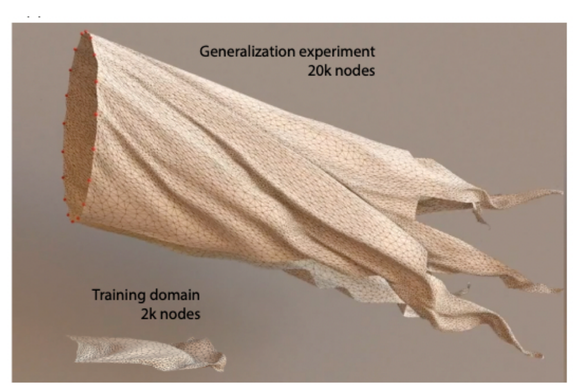
Typical methods are classical adaptive mesh refinement (AMR) solvers or deep learning-based surrogate models.

Deep learning-based surrogate models

Pros: offer speedup via larger spatial/temporal intervals and explicit forward

However,

- Most models only learn evolution with fixed mesh/grid topology, without optimizing the spatial resolution
- MeshGraphNets [1] use supervised learning to learn remeshing, limited by the remeshing provided by AMR solver
- RLAMR [2] uses RL to learn remeshing for FEM, but its goal is to reduce error only, and requires solver in the loop.

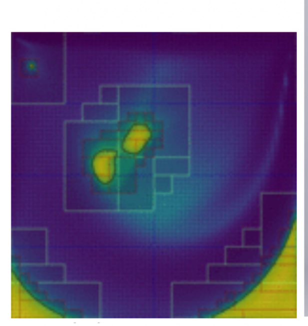


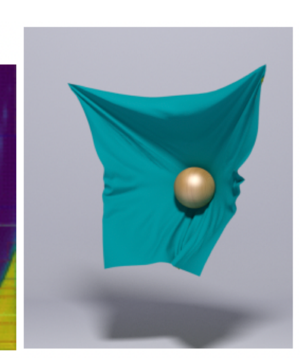
[1] Pfaff et al. 2020, [2] Yang et al, 2021

Classical AMR solvers

With some heuristic based on local state variation such as curvature, it adaptively refines or coarsens the mesh resolution.

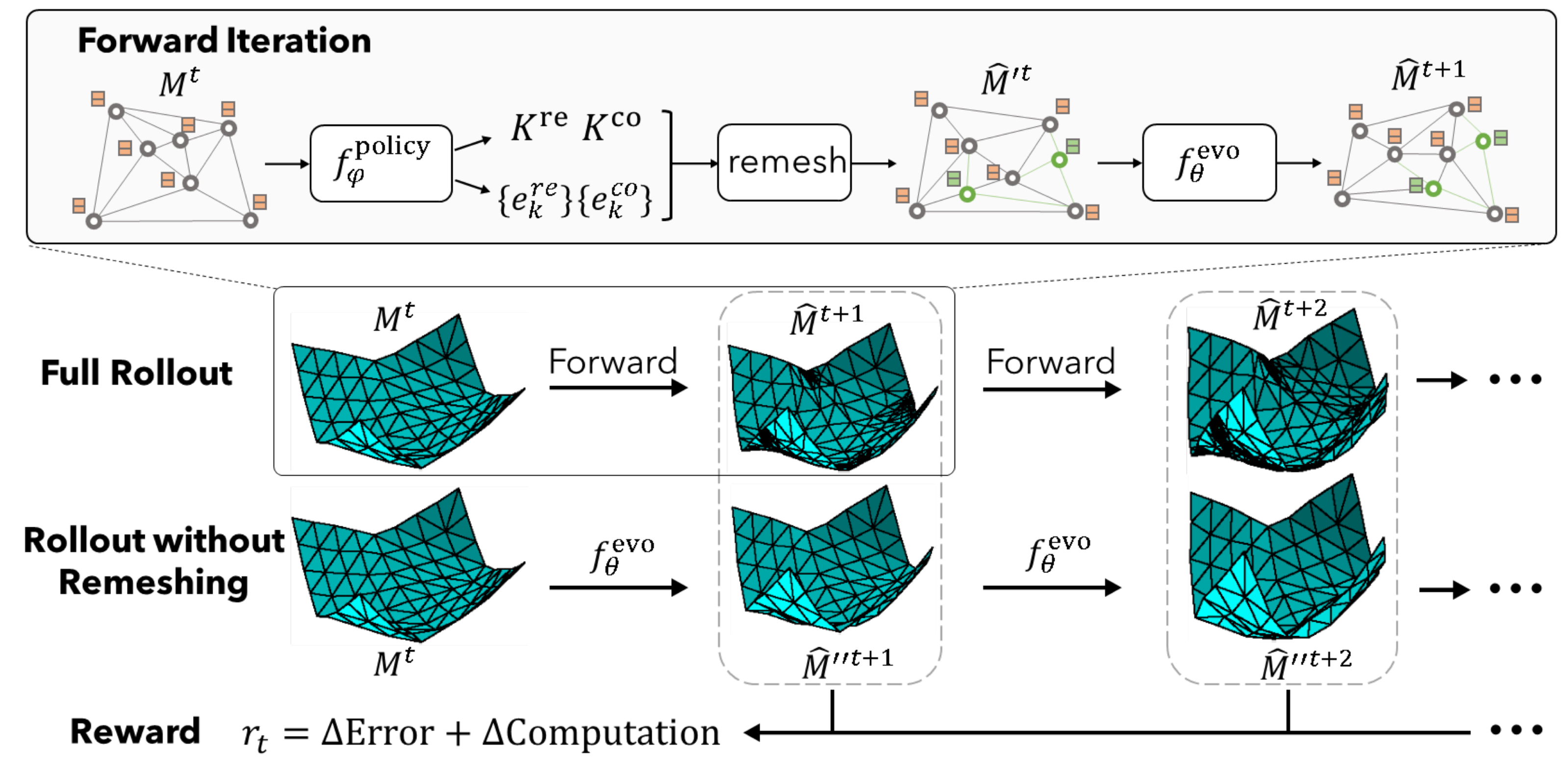
Challenge: Slow simulation





Approach

We Introduce the first fully DL-based surrogate model that jointly learns the evolution model and optimizes appropriate spatial resolution (LAMP).



LAMP consists of GNN-based evolution model for learning the forward evolution, and a GNN-based actor-critic for learning the policy of discrete actions of local refinement and coarsening of the spatial mesh.

Learning objective

Learning evolution: evolution model optimized to reduce the multi-step error

$$L^{\text{evo}} = L_S^{\text{evo}}[\mathbb{I}, f_\theta^{\text{evo}};\ \hat{M}^t] + L_S^{\text{evo}}\left[f_\varphi^{\text{policy}}, f_\theta^{\text{evo}};\ \hat{M}^t\right]$$

$$= \sum_{s=1}^{S} \alpha_s^{\text{policy}} l(\hat{M}^{t+s}, M^{t+s}) + \sum_{s=1}^{S} \alpha_s^{\mathbb{I}} l(\hat{M}''^{t+s}, M^{t+s})$$

Learning policy: the policy network learns to update the spatial resolution

$$r^t = (1-\beta)\cdot\Delta\text{Error} + \beta\cdot\Delta\text{Computation}$$

$$\begin{cases} \Delta\text{Error} = L_S^{\text{evo}}[\mathbb{I}, f_\theta^{\text{evo}};\ \hat{M}^t] - L_S^{\text{evo}}\left[f_\varphi^{\text{policy}}, f_\theta^{\text{evo}};\ \hat{M}^t\right] \\ \Delta\text{Computation} = C_S^{\text{evo}}[\mathbb{I}, f_\theta^{\text{evo}};\ \hat{M}^t] - C_S^{\text{evo}}\left[f_\varphi^{\text{policy}}, f_\theta^{\text{evo}};\ \hat{M}^t\right] \end{cases}$$

Action representation

Combination of a pair of integers specifying number of edges to split/coarsen and a set of edges to split/coarsen

$$a^t = (K^{\text{re}}, e_1^{\text{re}}, e_2^{\text{re}}, \ldots, e_{K^{\text{re}}}^{\text{re}}, K^{\text{co}}, e_1^{\text{co}}, e_2^{\text{co}}, \ldots, e_{K^{\text{co}}}^{\text{co}})$$

The log-probability for the sampled action a^t is given by

$$\log p_\varphi(a^t|M^t) = \underbrace{\log p_\varphi(K^{\text{re}}|M^t) + \sum_{k=1}^{K^{\text{re}}} \log p_\varphi(e_k^{\text{re}}|M^t)}_{\text{Split action}}$$

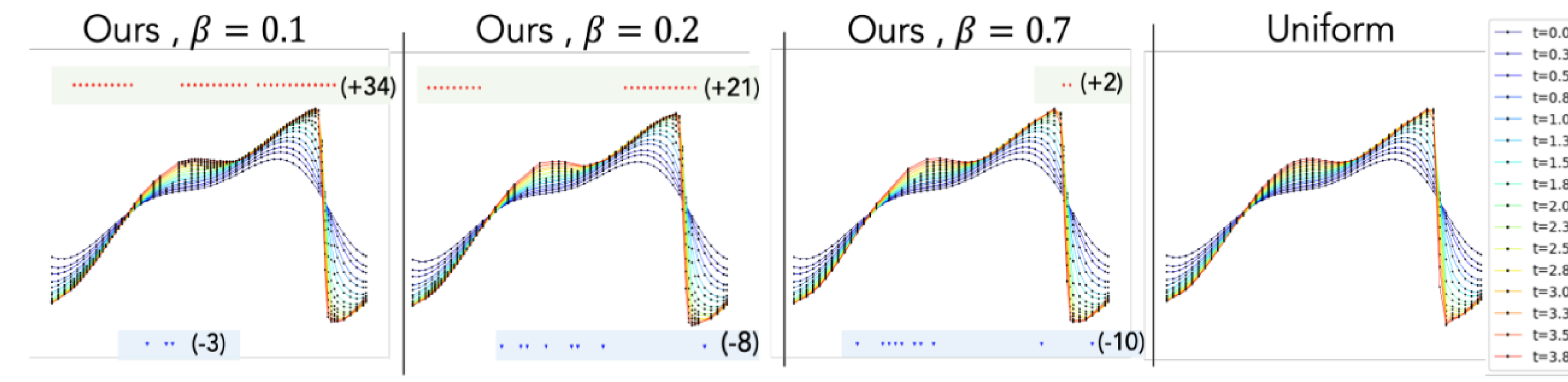
$$+ \underbrace{\log p_\varphi(K^{\text{co}}|M^t) + \sum_{k=1}^{K^{\text{co}}} \log p_\varphi(e_k^{\text{co}}|M^t)}_{\text{Coarsen action}}$$

Result

LAMP outperforms state-of-the-art deep learning surrogate models with up to 39.3% error reduction, and is able to adaptively trade-off computation to improve long-term prediction error.

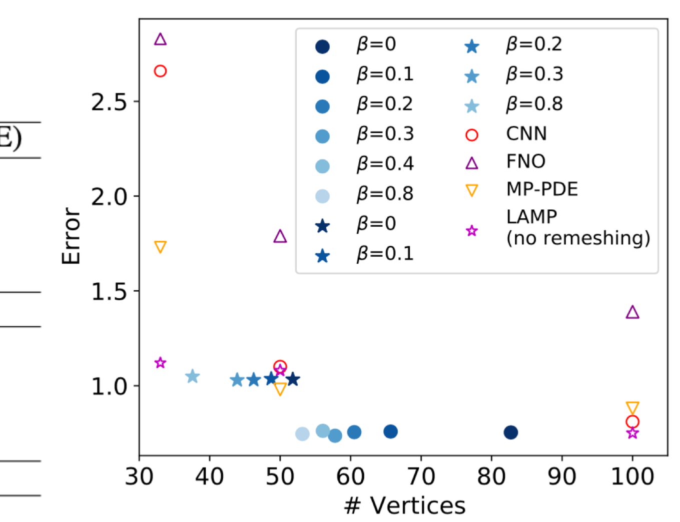
1D simulation

With larger β, LAMP coarsens more and refines less, and focuses the refinement on the most dynamic region:

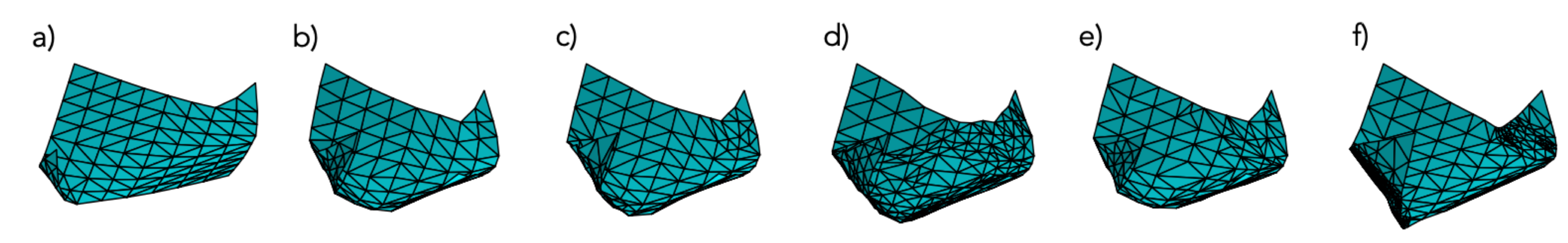


LAMP improves the Pareto frontier for Error vs. Computation:

Model	Initial # vertices	Average # vertices	Error (MSE)
CNN	33	33.0	2.66
FNO	33	33.0	2.83
MP-PDE	33	33.0	1.73
LAMP (no remeshing)	33	33.0	1.12
LAMP (ours)	33	37.6	**1.05**
CNN	50	50.0	1.10
FNO	50	50.0	1.79
MP-PDE	50	50.0	0.98
LAMP (no remeshing)	50	50.0	1.08
LAMP (ours)	50	53.2	**0.75**



2D mesh-based simulation


a) Initial state
b) LAMP (no remeshing)
c) MeshGraphNets + GT remeshing
d) MeshGraphNets + heuristic remeshing
e) LAMP (ours)
f) Ground truth

LAMP outperforms MeshGraphNets with ground-truth remeshing, and heuristic remeshing:

Model	Initial # vertices	Average # vertices	Error (MSE)
MeshGraphNets + GT remeshing	102.9	115.9	5.91e-4
MeshGraphNets + heuristics remeshing	102.9	191.9	6.38e-4
LAMP (no remeshing)	102.9	102.9	6.13e-4
LAMP (ours)	102.9	123.1	**5.80e-4**

Discussion

Takeaway: by jointly learning the evolution model and learning to optimize the spatial resolution, LAMP

- achieves a controllable tradeoff between error and computational cost
- Improves the Pareto frontier of error vs. computational cost.

For more details, see our paper by scanning QR code: