2021/11/30
International Workshop on the Integration of (Simulation + Data + Learning)
Towards Society h3-Open-BDEC

Enhancement of Molecular Dynamics Simulation by Machine Learning

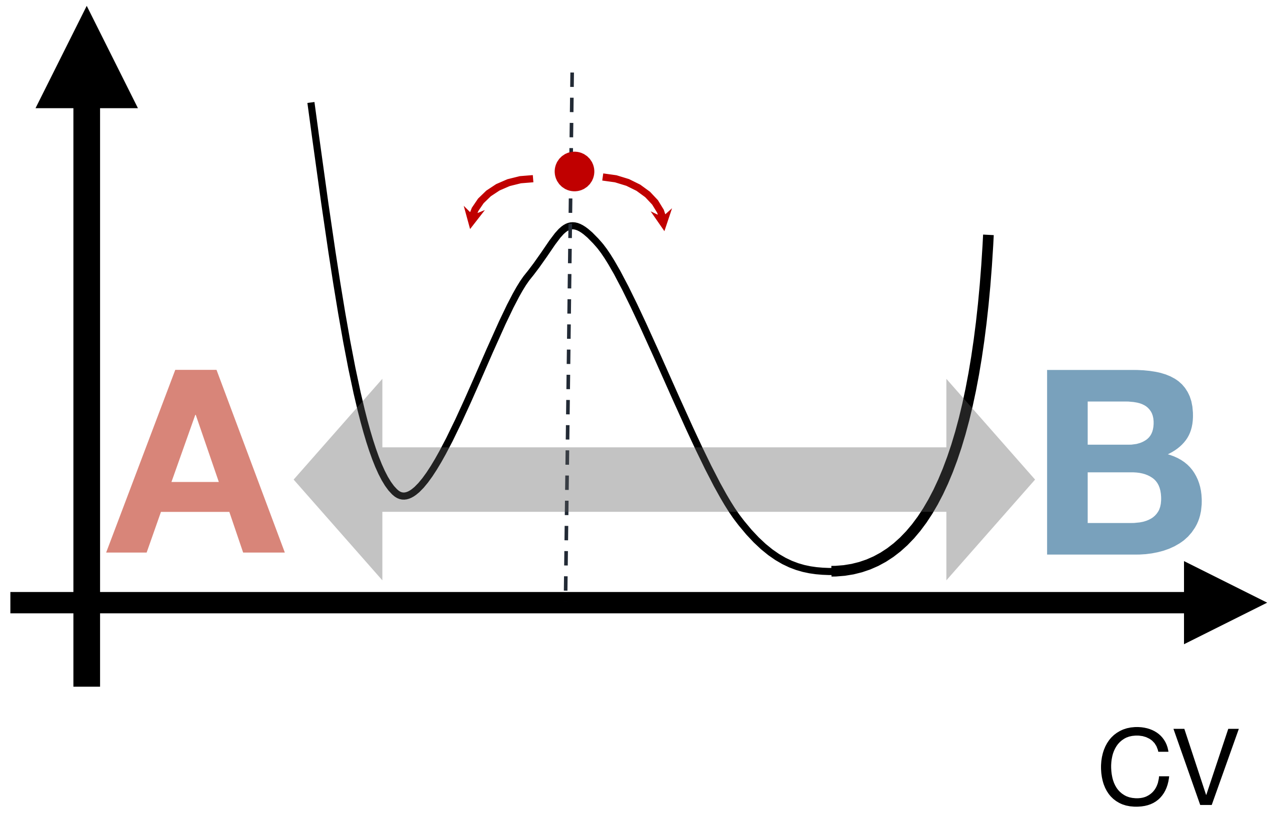
Hayato SHIBA
Supercomputing Research Division
Information Technology Center, University of Tokyo

Joint research with Takashi Shimokawabe

Towards "long-time" molecular dynamics

(> milliseconds)

— Rare events = phenomena that takes extremely long time due to potential energy barriers
ex) allosteric events, nucleation, crystallization, fracture, …



— Due to the approach of Moore's law, direct MD simulations for $>10^{10}$ steps ($\rightleftarrows$ μs) will remain difficult for coming years.

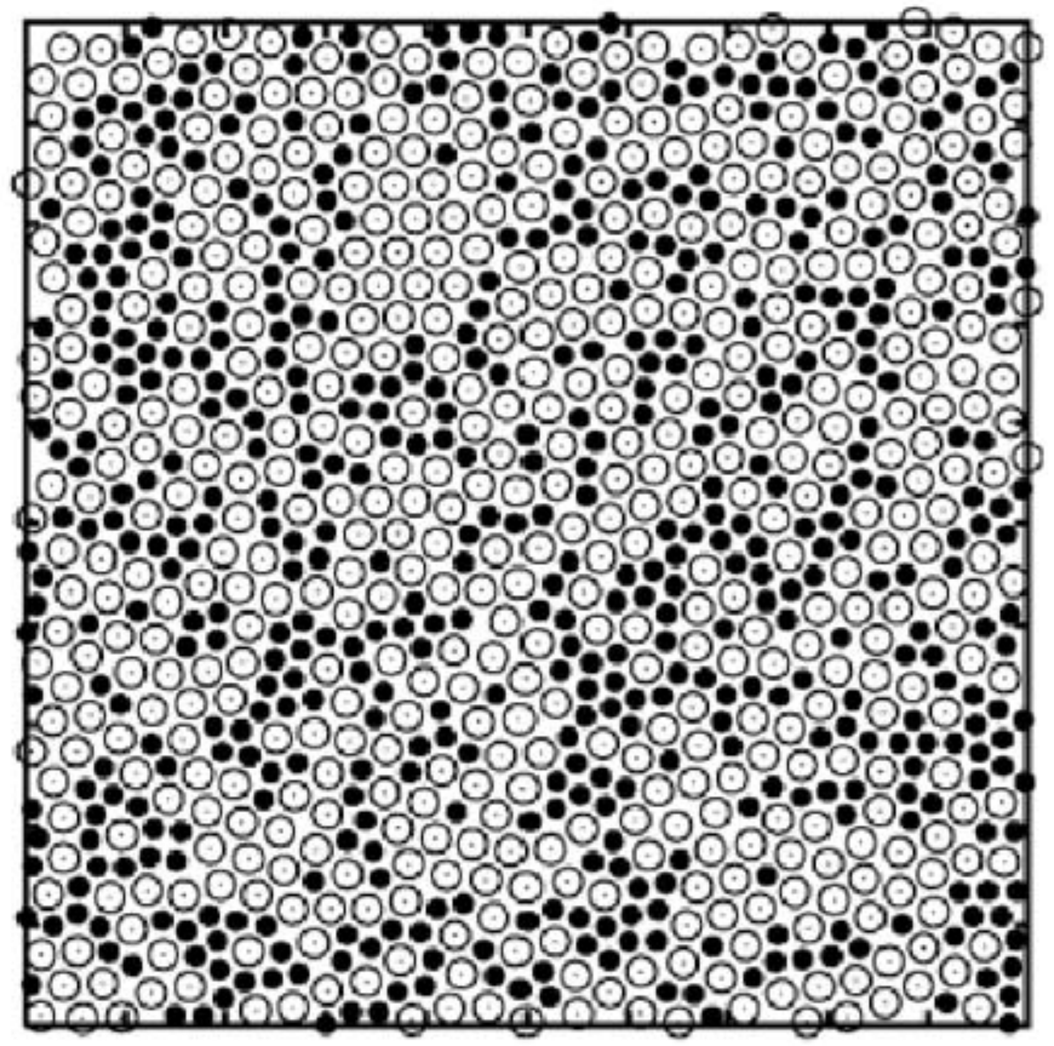
— Biased simulation / enhanced sampling can enforce the system to overcome this energy barrier by additional constraints. However,

- — we need to know advance a small number of good collective variables (CVs) that properly describing the energy landscape (energy surface).
- — It can only describe dynamical paths between a small number of energy minima.

— For many types of complex & time-dependent molecular dynamics, different approaches may be needed.

Glasses

(2D schematic)



- Formed by rapidly cooling a liquid
- keep random structure $\simeq$ liquid
- particle motion is frozen $\simeq$ solid

⬇

glasses
= *liquids with diverging timescale*

3D Kob-Andersen LJ model

$$v_{\alpha\beta}(r) = 4\epsilon_{\alpha\beta}\left[\left(\frac{\sigma_{\alpha\beta}}{r}\right)^{12} - \left(\frac{\sigma_{\alpha\beta}}{r}\right)^{6}\right]$$

$$\epsilon_{22} = 0.5\epsilon_{11},\ \epsilon_{12} = 1.5\epsilon_{11}$$

$$\sigma_{22} = 0.8\sigma_{11},\ \sigma_{12} = 0.88\sigma_{11}$$

with the composition **80**(1)**:20**(2)

There are little in structures, the DYNAMICS mainly matter

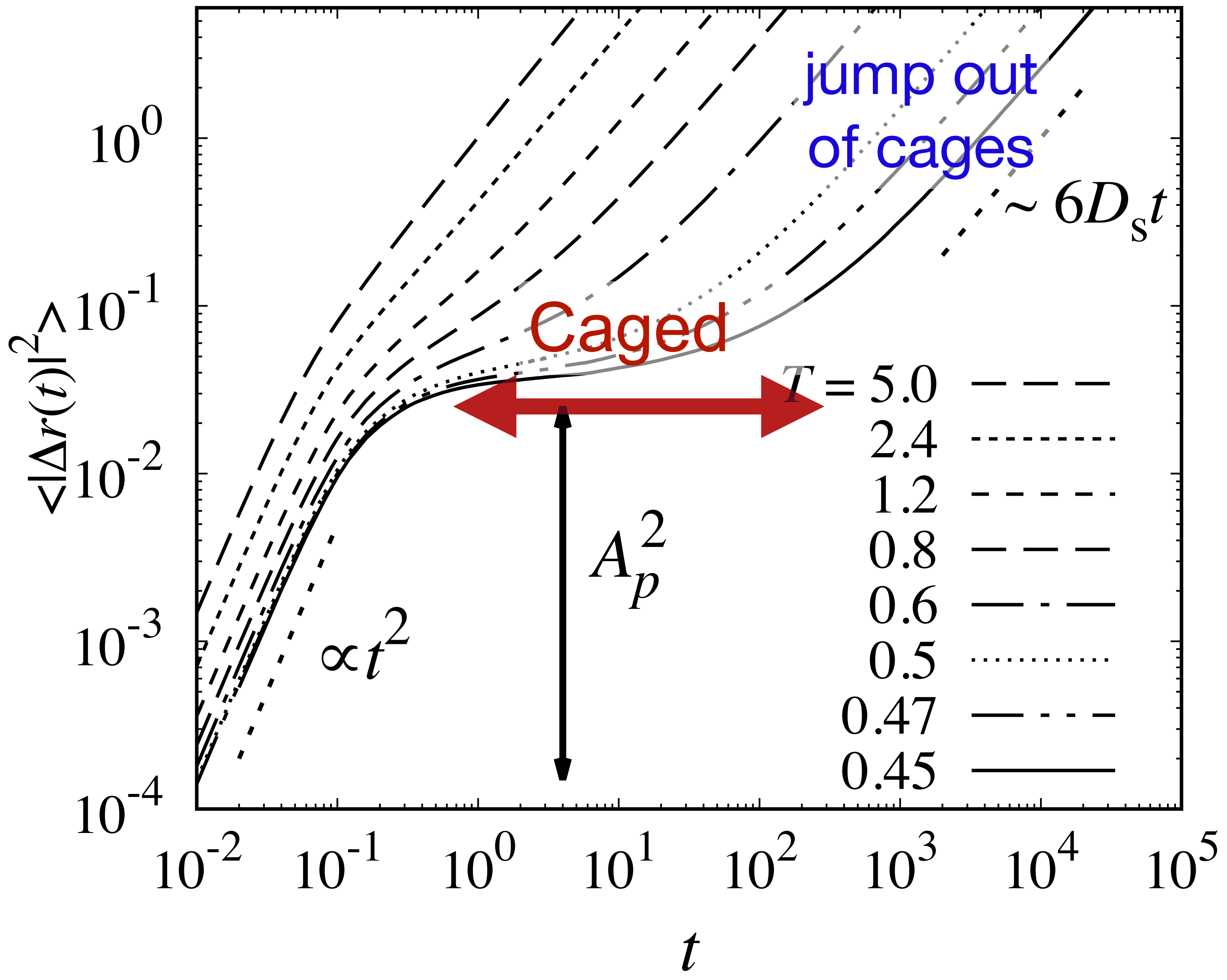
✓ MD simulation is a strong tool

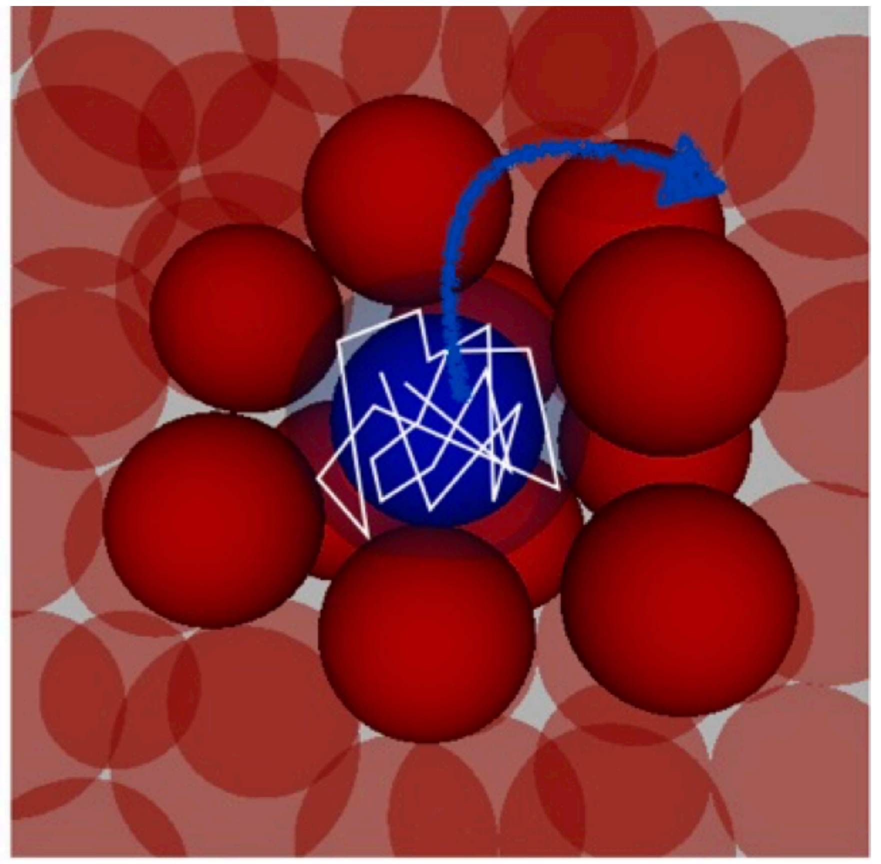
- slow dynamics = *intermittent* jumps
- for most of time, particles are just vibrating

(we may want to skip by AI)

- *nevertheless* the dynamics is heterogeneous

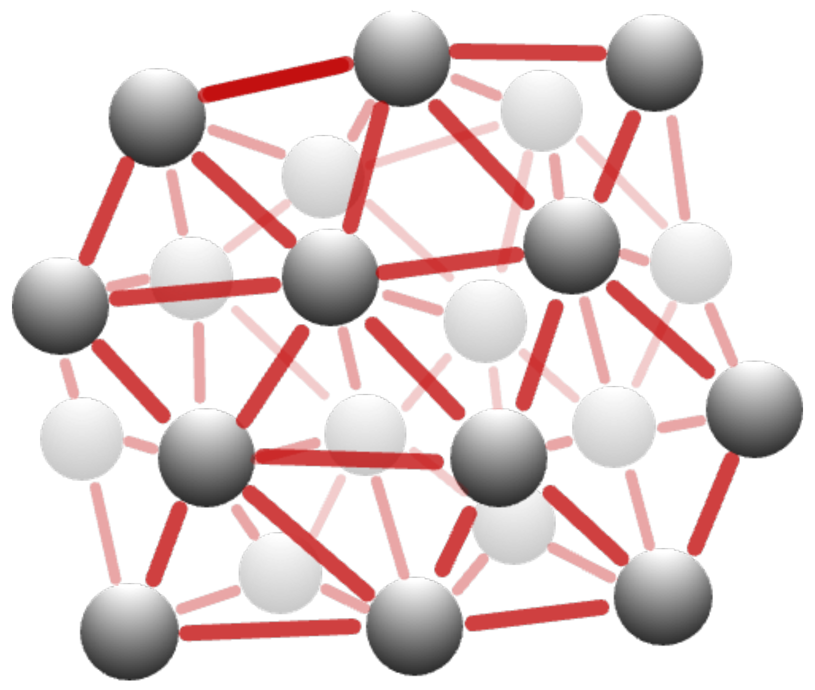
MSD = (particle traveling distance)^2





GNN architechture

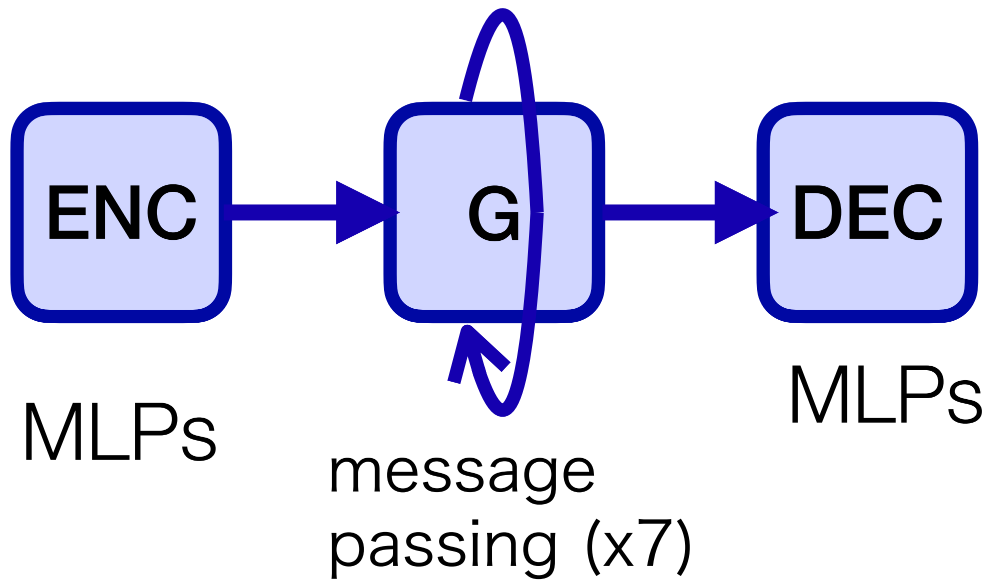
$N = 4096$

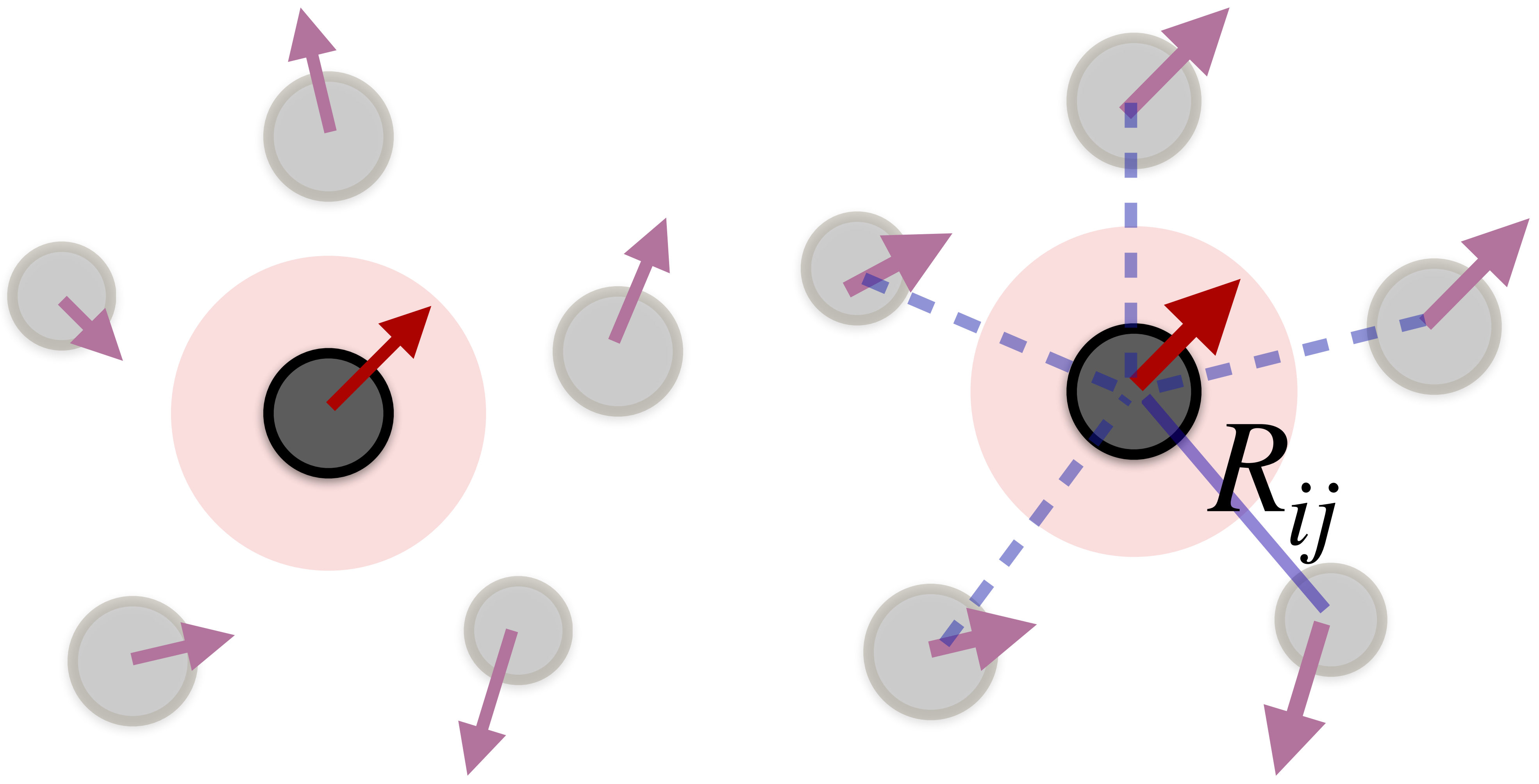


	ENCODER	DECODER (DeepMind,2020)	DECODER **(Our work)**
nodes #=4096	particle type (1 or 2)	**propensity** $\langle \vert \mathbf{r}_i(t+\Delta t) - \mathbf{r}_i(t) \vert \rangle$	
edges #~170000	relative position ($\boldsymbol{R}_{ij}$)		**neighbor distance change** $\langle R_{ij}(t+\Delta t) - R_{ij}(t) \rangle$

node = particle
edge = nearby pairs $(R_{ij} < 2.0\sigma_{\mathrm{AA}})$

on each node/edge





- *propensity*
  = self motion of one particle
- *neighbor distance change*
  = **larger # of data**
  structural changeover

  some physics may be informed

Dataset

We use the dataset distributed by DeepMind group.

— prepare 400 indep. particle configurations ($N = 4{,}096$) by annealing

— run 30 indep. velocities from each (isoconfigurational ensemble)

→ 12,000 simulations

in glasses, propensity tends to be determined by the velocities
and not depend strongly on the velocities.

— data augmentation by 24 cubic rotation at random

— compute and store the particle motion after long-time MD.

Learning

Source code

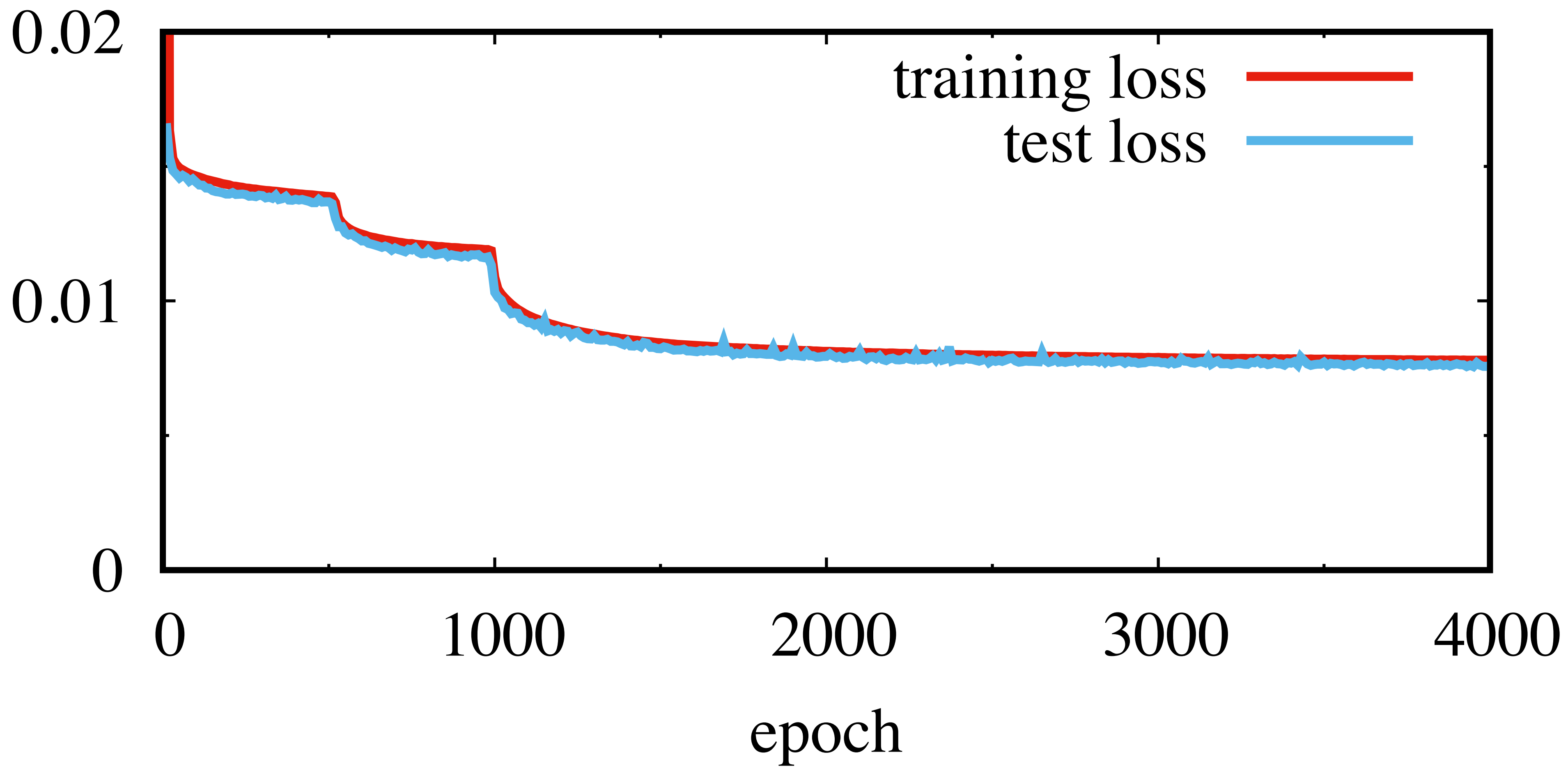
https://github.com/deepmind/deepmind-research

@Google Research

JAX = [Automatic differentiation]
+ [XLA compiling of NumPy for GPUs]

additional packages required for ML

GNN = jraph + dm-haiku, optimizer = optax



wrapping/decoration by **`@jax.jit`**

→ speed up by just-in-time XLA compiling

Time per epoch (learning for 400 configs.)

measured on Wisteria-A, data I/O not included

1 CPU core	3240 s / epoch
1 GPU	168 s / epoch
1 GPU @jax.jit (XLA compilation)	8.5 s / epoch

1 GPU = NVIDIA A100 SVMe 40GB

data parallel not yet test for technical reasons

```
@jax.jit
def loss_fn(params, graph, targets, mask):
  decoded_nodes = network_apply(params, graph) * mask
  return (jnp.sum((decoded_nodes - targets)**2 * mask) / jnp.sum(mask))

@jax.jit
def update(params, opt_state, graph, targets, mask):
  loss, grads = jax.value_and_grad(loss_fn)(params, graph, targets, mask)
  updates, opt_state = opt_update(grads, opt_state)
  return optax.apply_updates(params, updates), opt_state, loss

train_stats = []
logging.info('Start training')
for epoch in range(n_epochs):
  logging.info('Start epoch %r', epoch)
  random.shuffle(training_data)
  for graph, targets, mask in training_data:
    params, opt_state, loss = update(params, opt_state, graph, targets, mask)
    train_stats.append(loss)
```

$T = 0.44, \quad$ prediction over 3.3×10^7 MD steps ($\simeq 3\tau_\alpha$)

in 2D cross sections

node learning

= particle propensity

(reproduction)

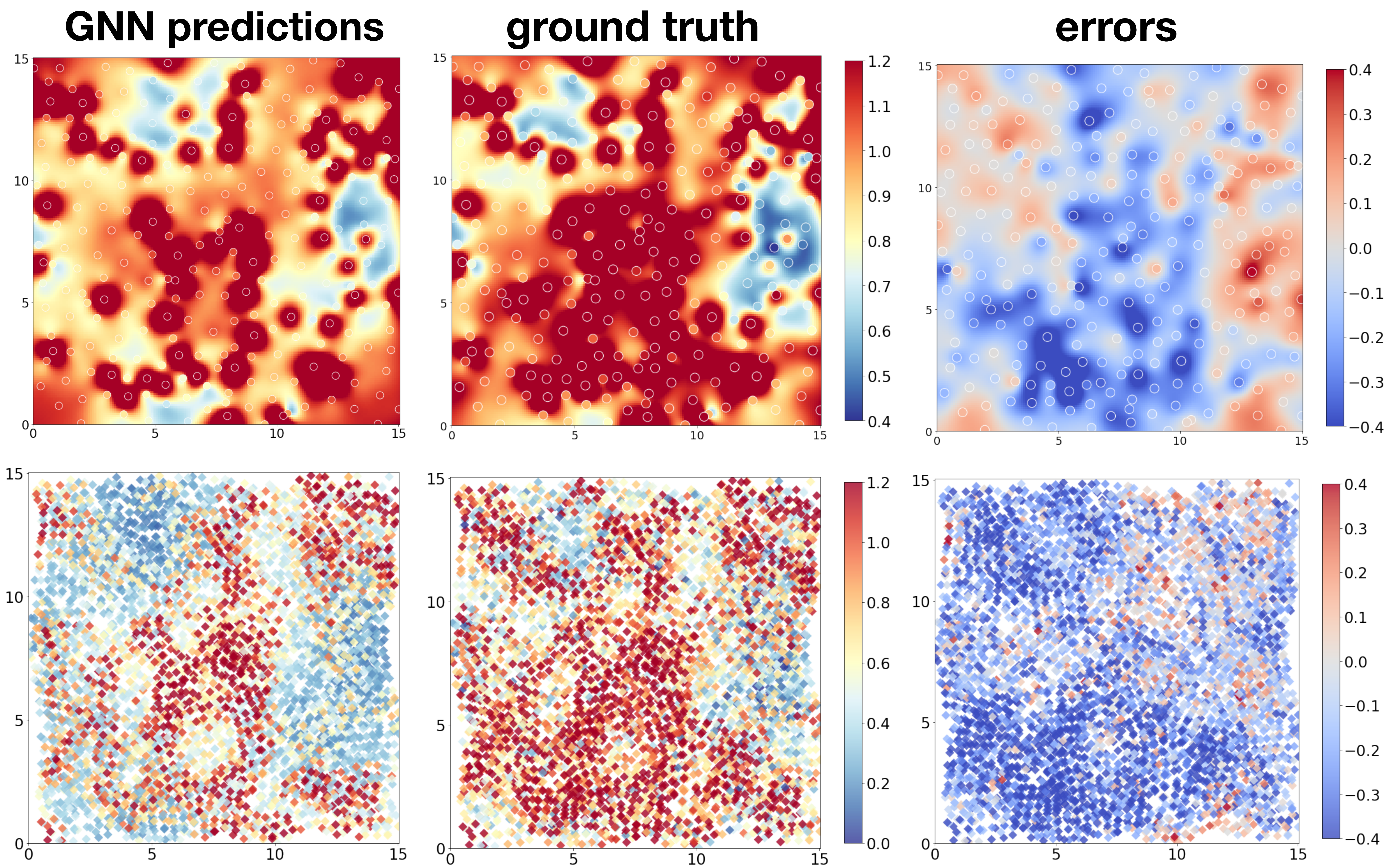
=4096

edge learning

= change in pair distance

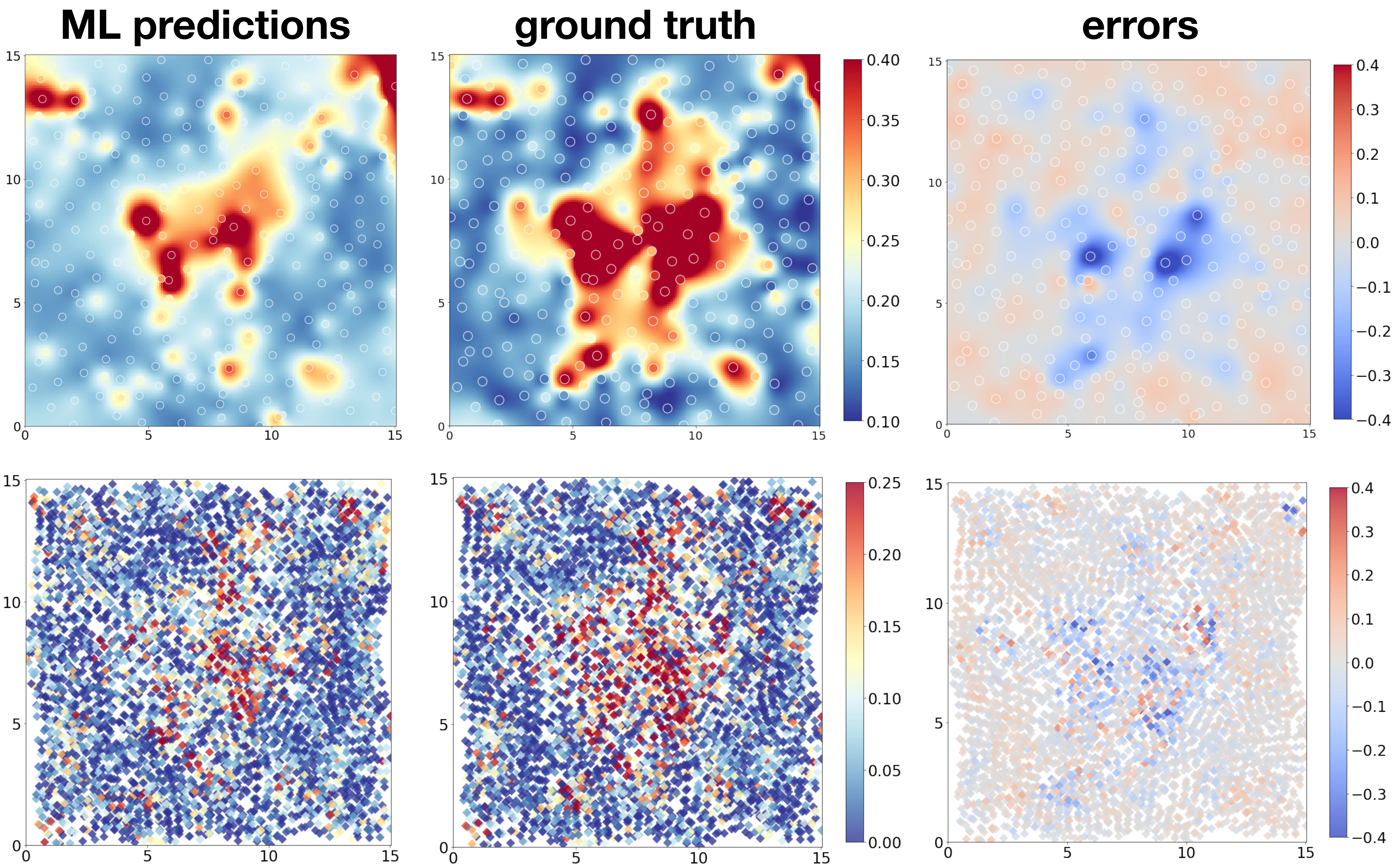
(this study)

~ 170000



$T = 0.44, \quad \text{prediction over } 4.5 \times 10^5 \text{ MD steps } (\simeq 0.1\tau_\alpha)$

in 2D cross sections



node propensity

Pearson correlation

$\rho = 0.597$

edge propensity

Pearson correlation

$\rho = 0.789$

“Hot spots"

Summary

✓ Power of GNNs for predicting the slow dynamics from **one** MD snap.

— based on a recent work of DeepMind group (2020)

— the long-time dynamics is surprisingly well predicted by learning changes in pair distances.

✓ We are stepping toward the "AI glass simulator".