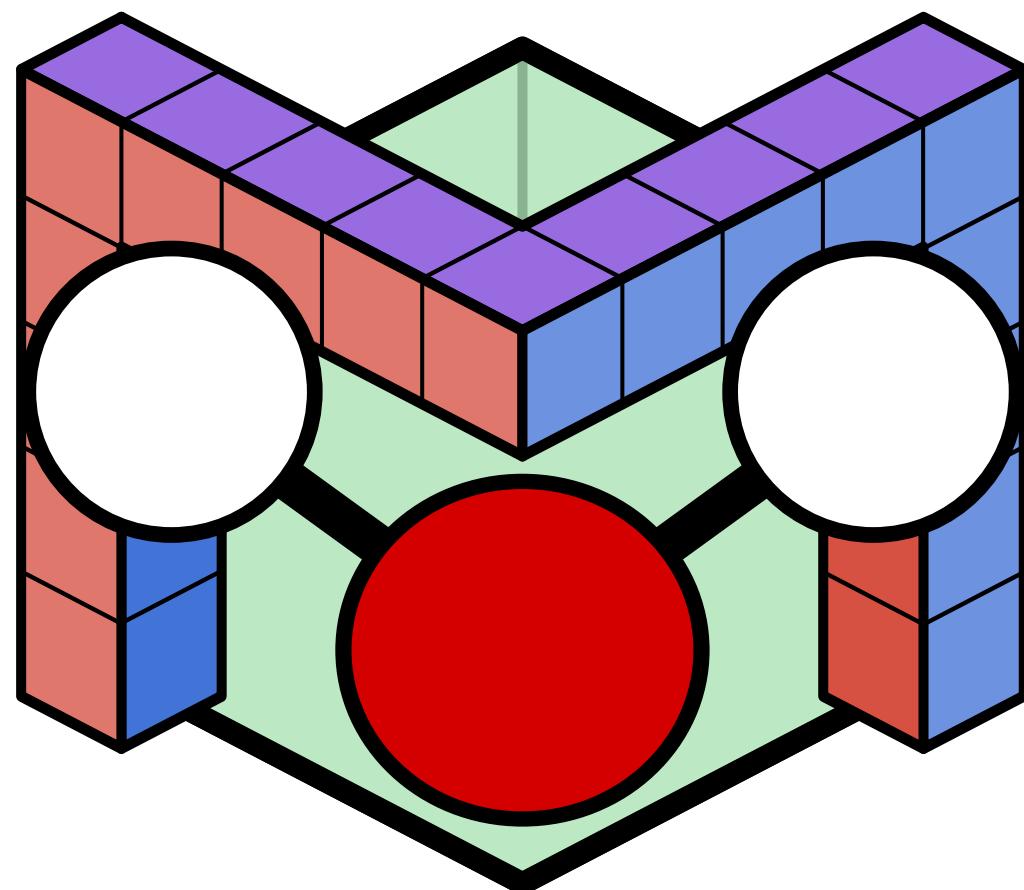


Building bridges between machine learning and simulation engines



metatomic



Guillaume Fraux



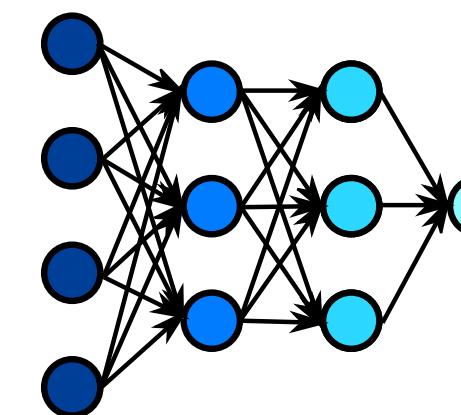
Download these slides

What are we doing?

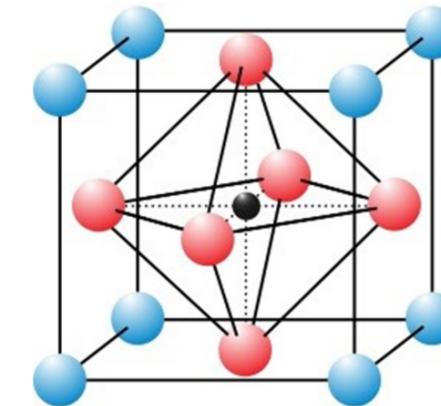
- Share atomistic machine learning models between developers and users



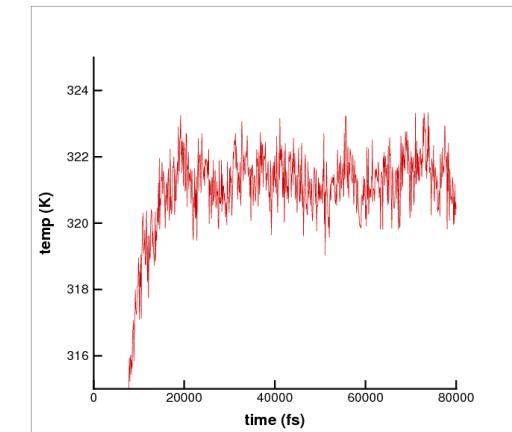
"I want to create new ML models"



"I want to use ML for research on my system"



"I want to use ML together with my special simulation tool"



Modular ecosystem to exchange ML models

What are we doing?

- Share atomistic machine learning models between developers and users
- Reduce the amount of work to interface ML models with simulation engines

DeepMD

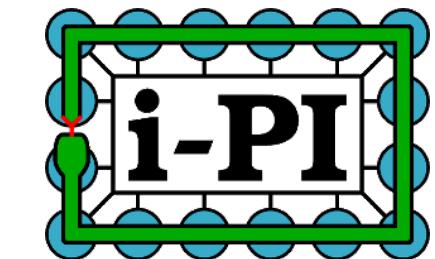
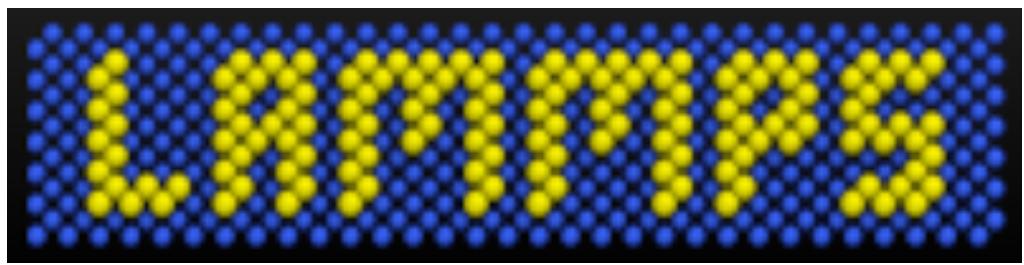
PET

ANI

MACE

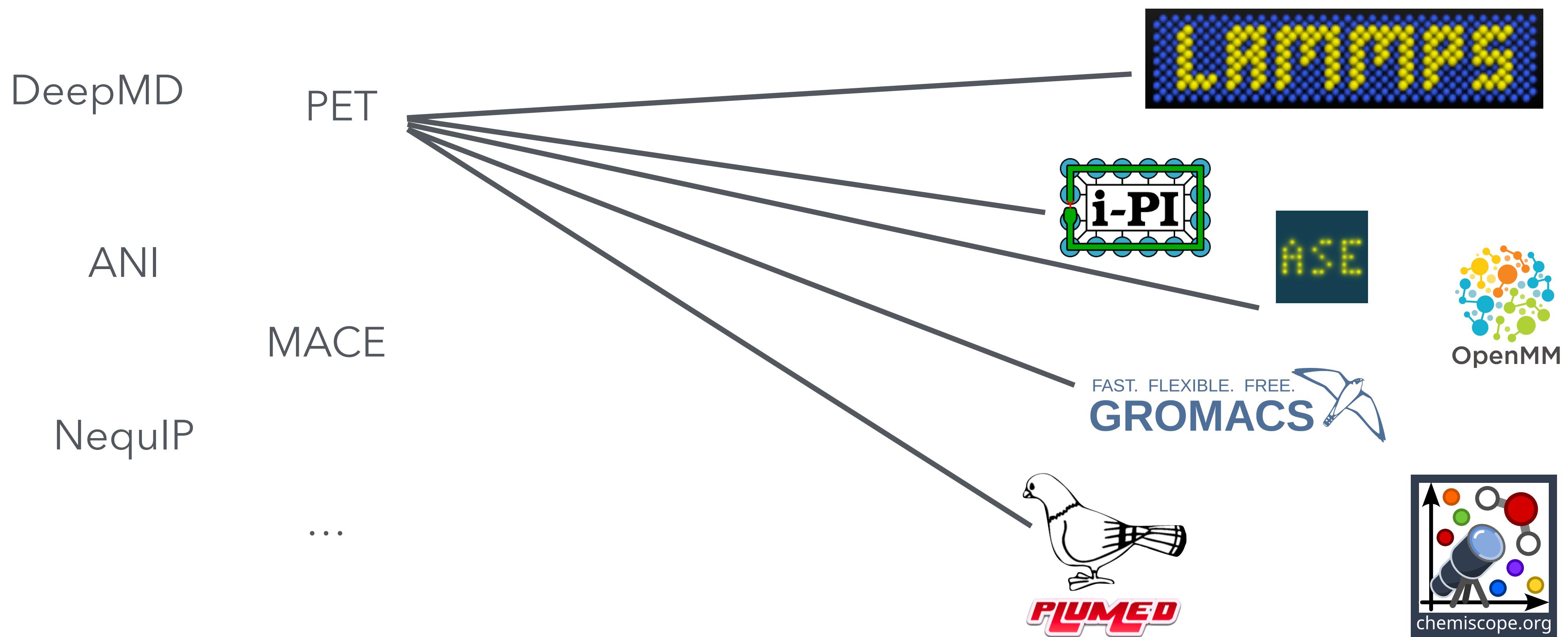
NeqIP

...



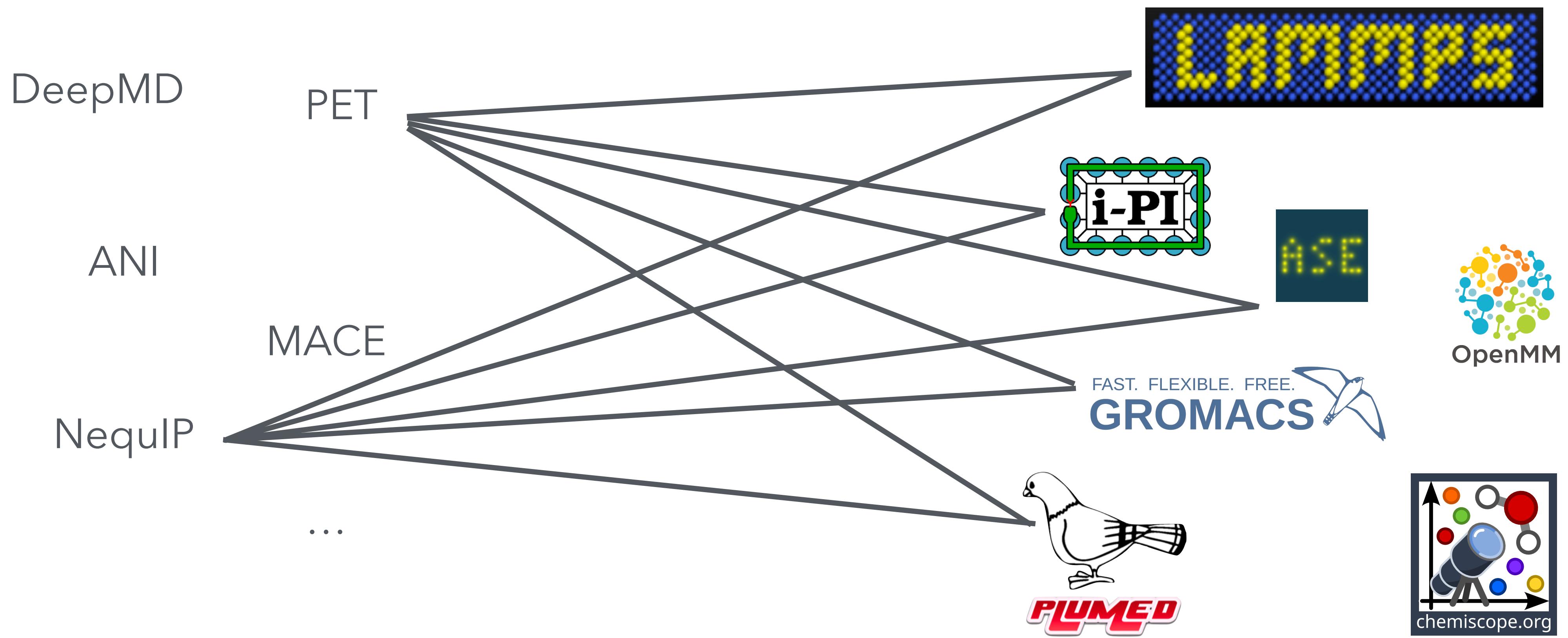
What are we doing?

- Share atomistic machine learning models between developers and users
- Reduce the amount of work to interface ML models with simulation engines



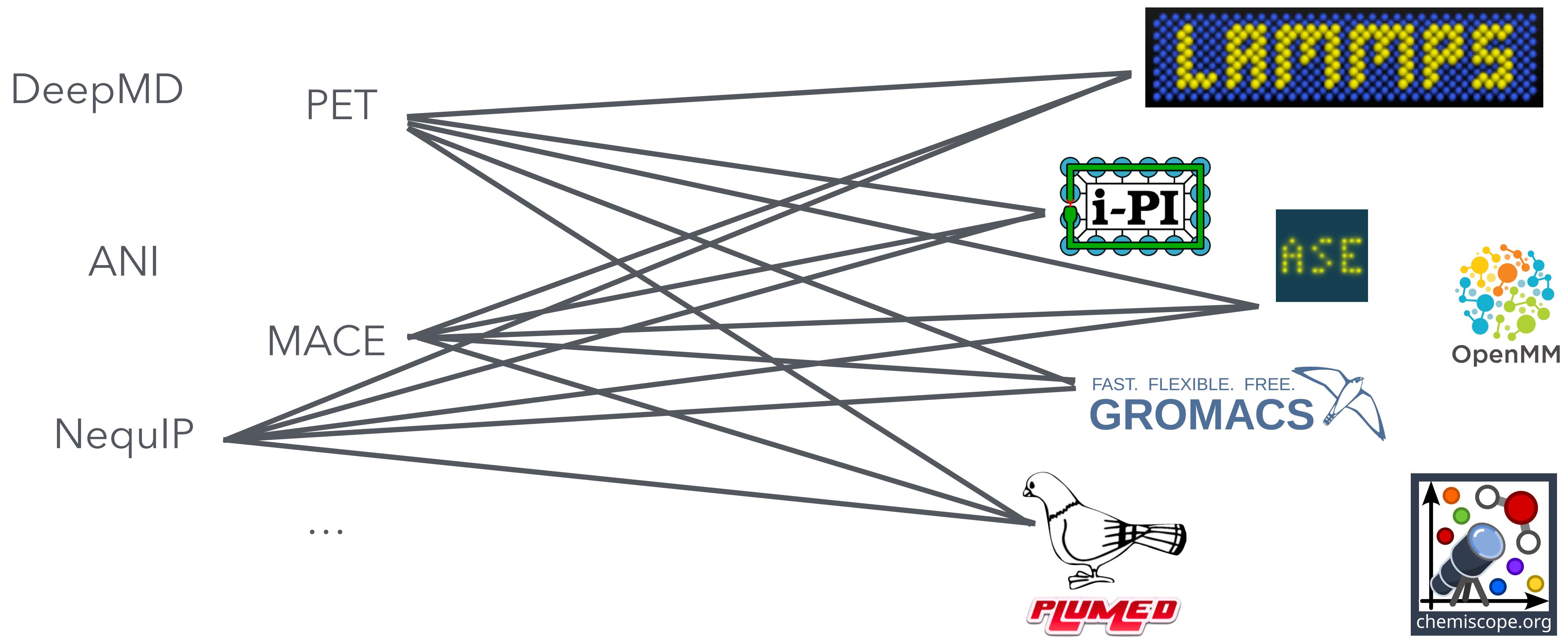
What are we doing?

- Share atomistic machine learning models between developers and users
- Reduce the amount of work to interface ML models with simulation engines



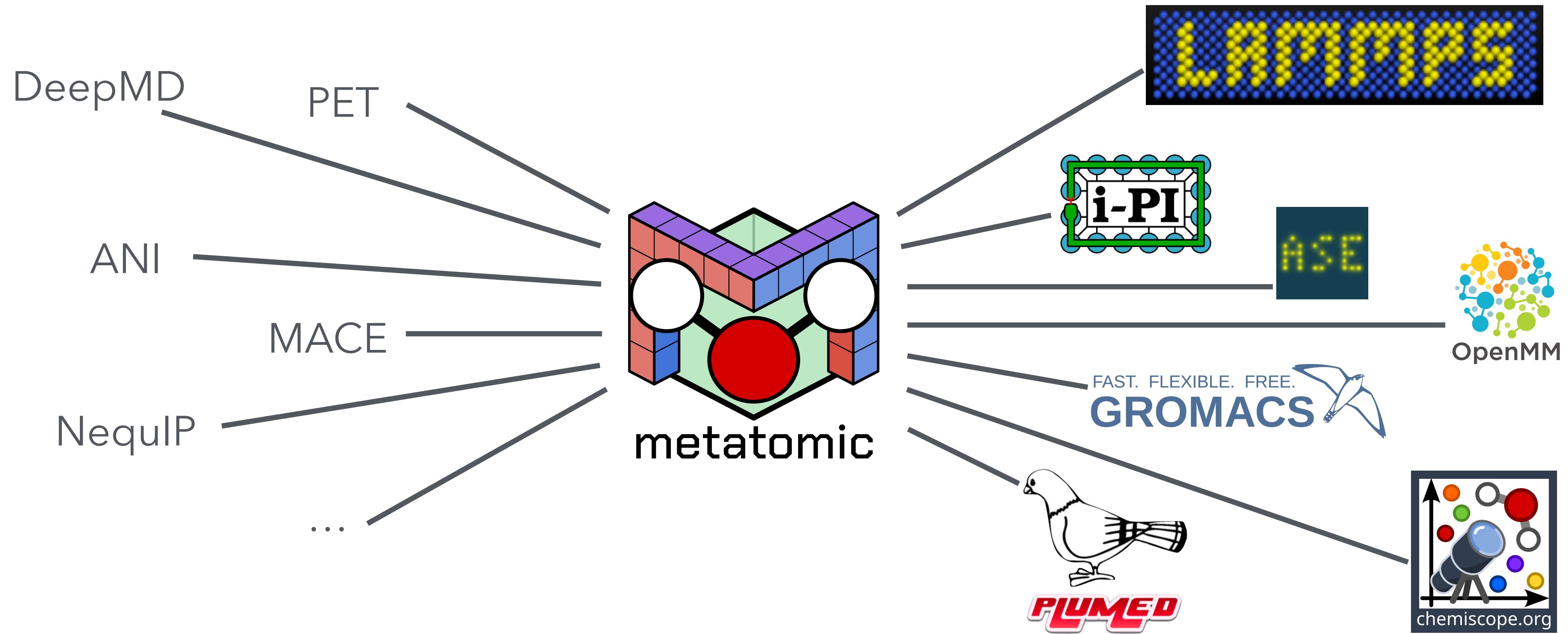
What are we doing?

- Share atomistic machine learning models between developers and users
- Reduce the amount of work to interface ML models with simulation engines



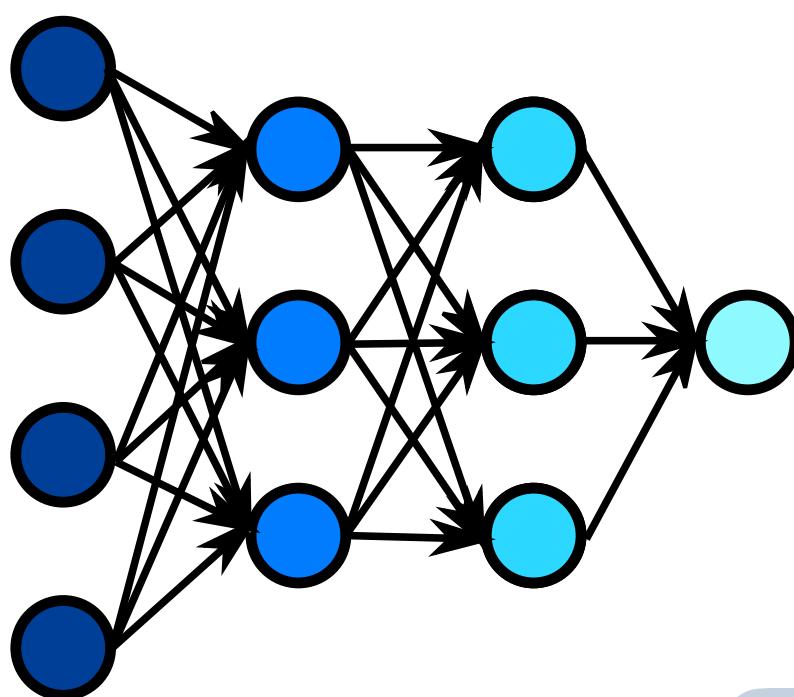
What are we doing?

- Share atomistic machine learning models between developers and users
- Reduce the amount of work to interface ML models with simulation engines



What are we doing?

- Share atomistic machine learning models between developers and users
- Reduce the amount of work to interface ML models with simulation engines
- Allow using ML models for every kind of tasks at the atomic scale



Run large-scale MD simulations

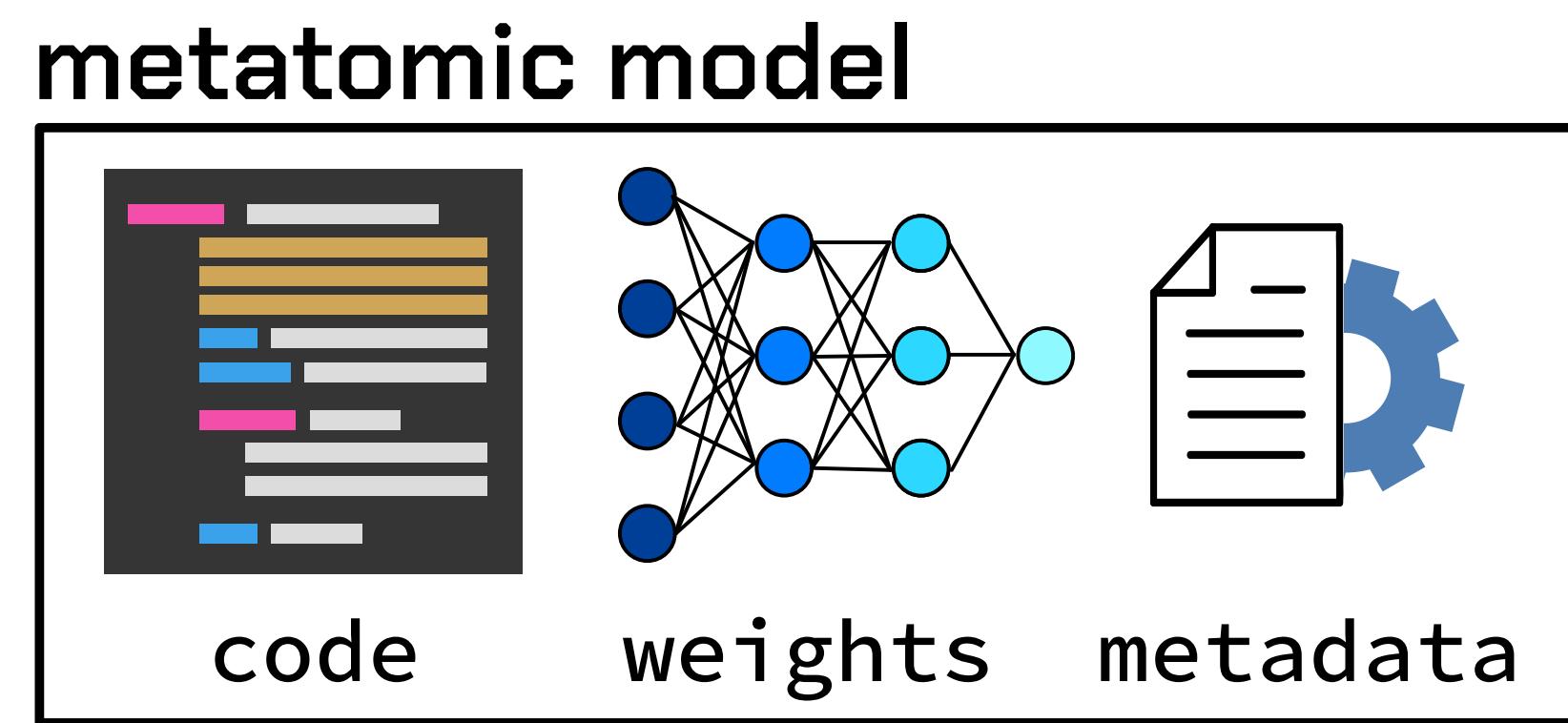
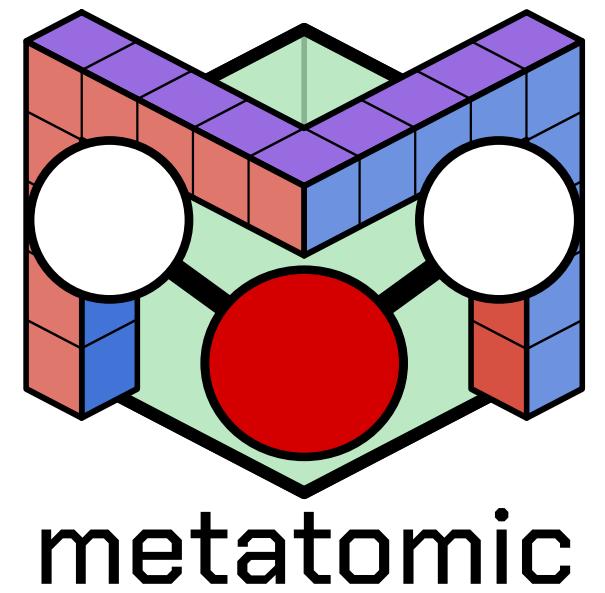
Quantify uncertainty

Predict atomic properties

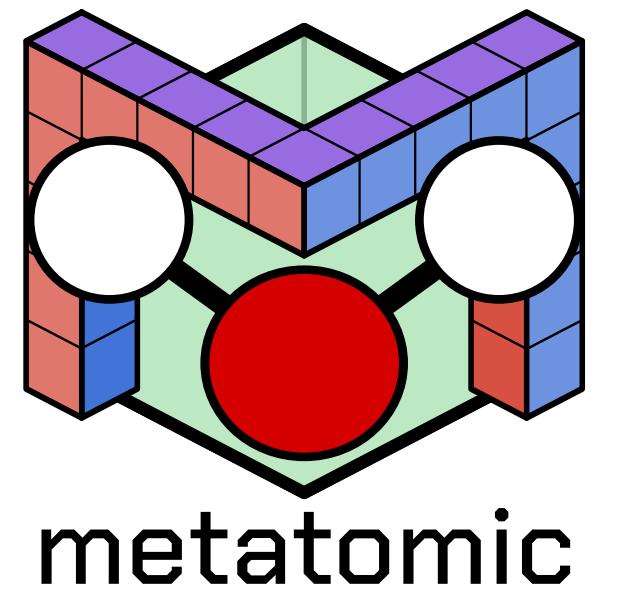
Use ML-learned collective variables

Integrate inside *ab initio* calculations

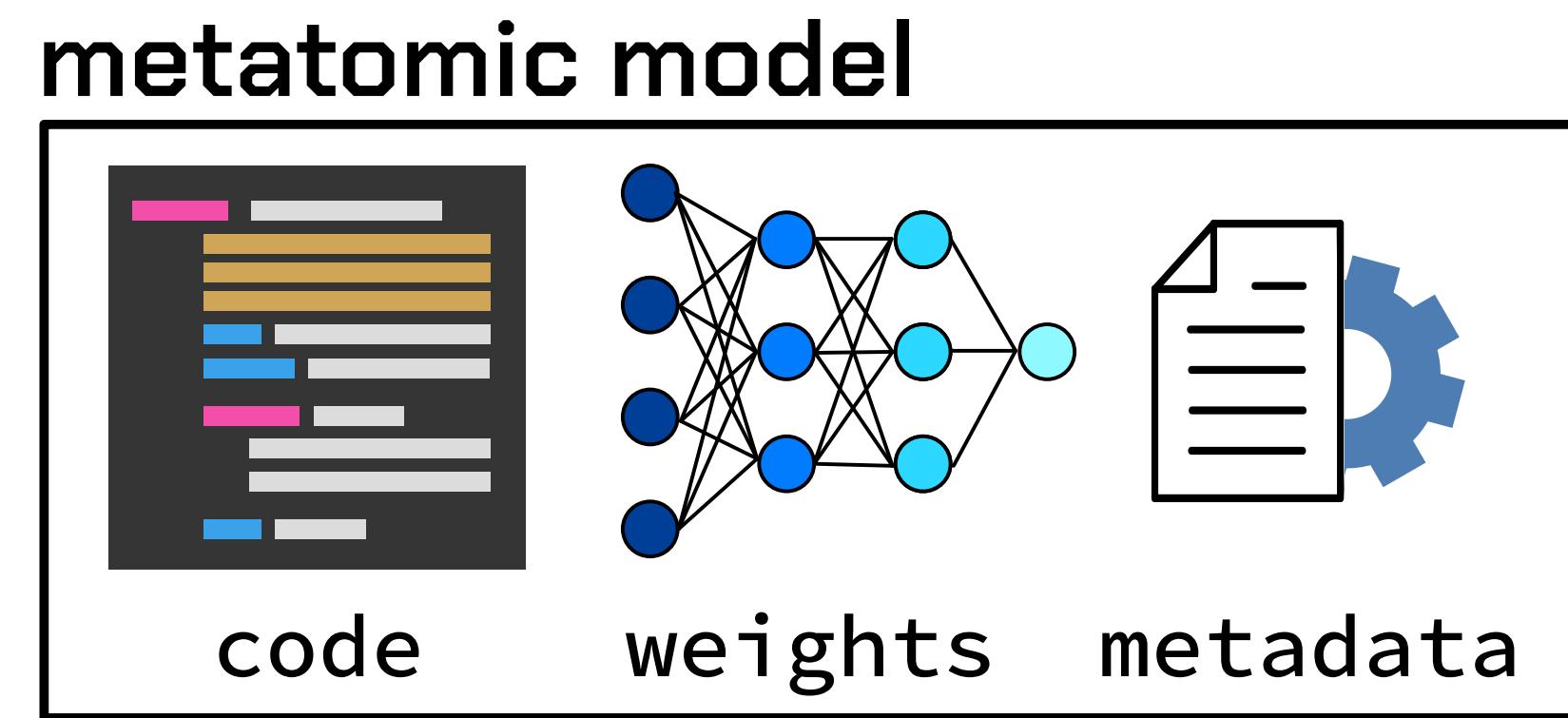
The metatomic API



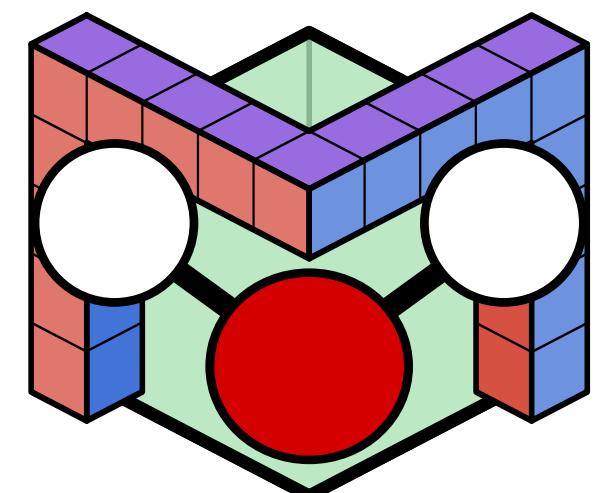
The metatomic API



- PyTorch/TorchScript model

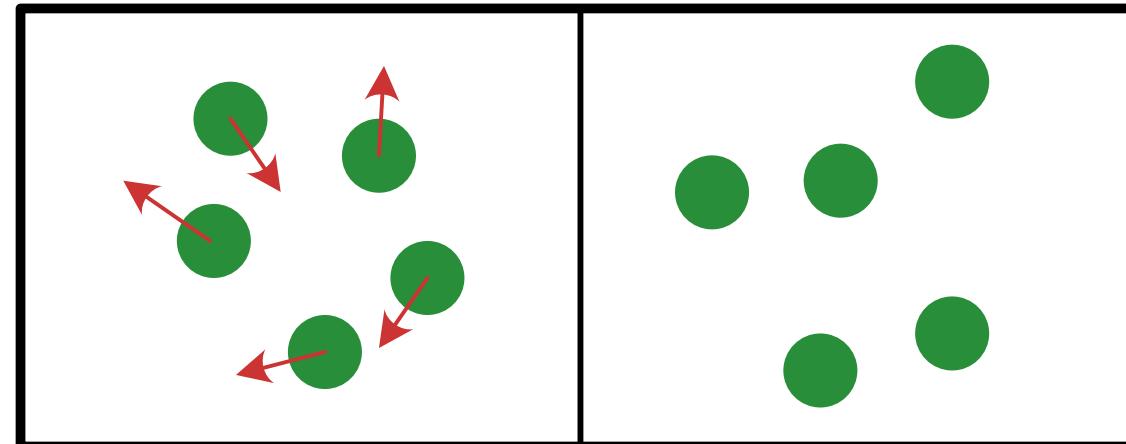


The metatomic API



metatomic

simulation engine

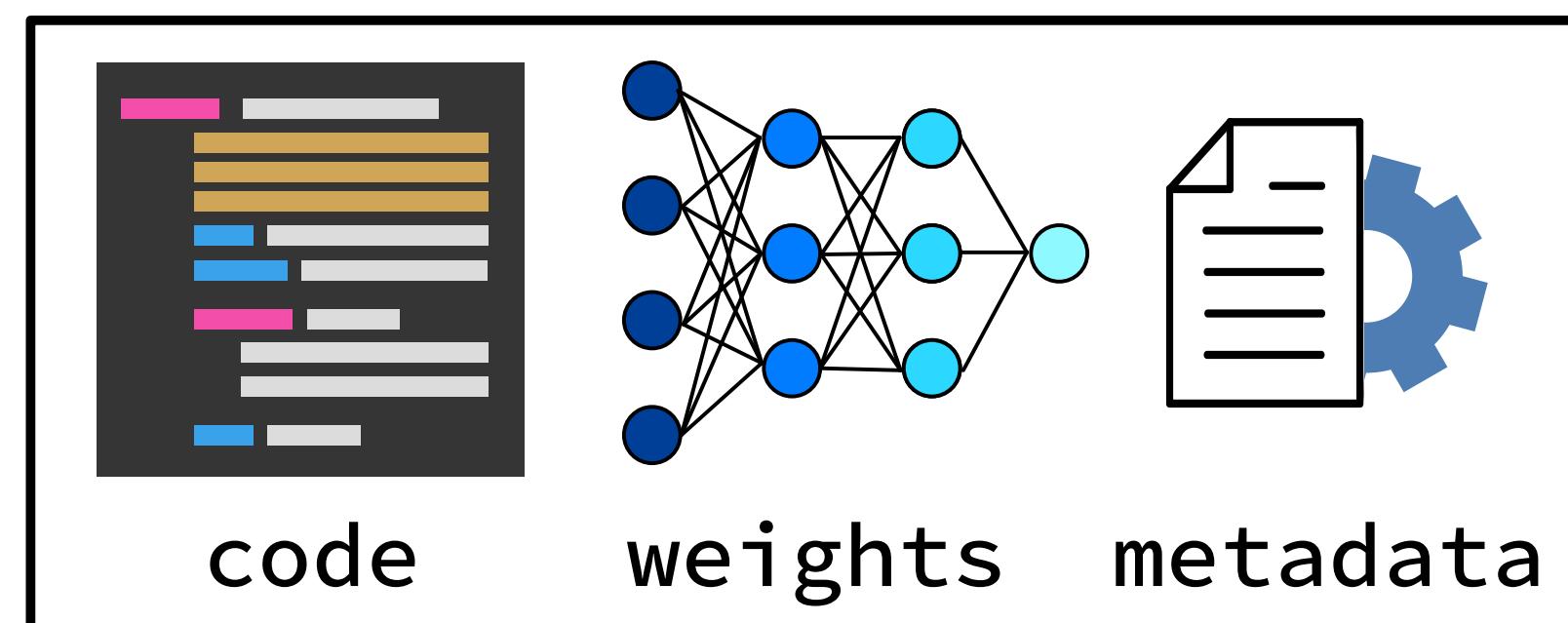


prepare

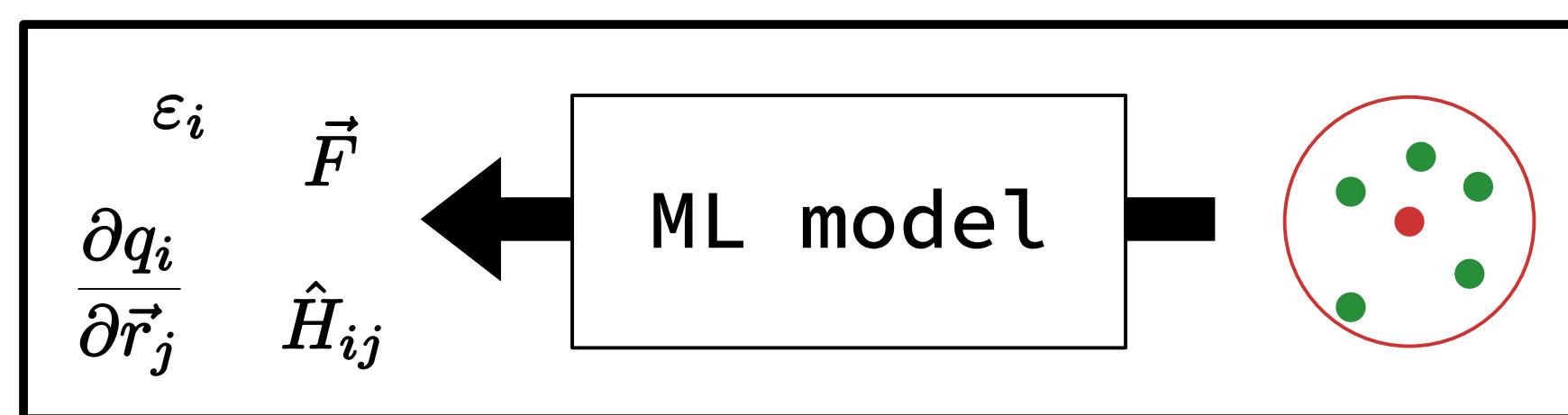


- PyTorch/TorchScript model
- Execute **any** model in **any** simulation engine, including large scale MPI tools

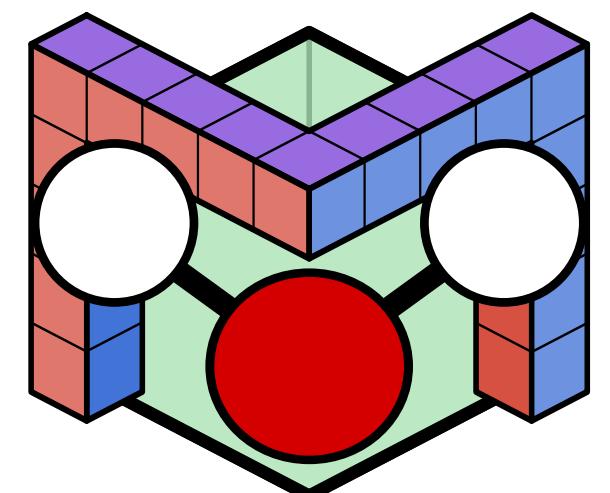
metatomic model



execute

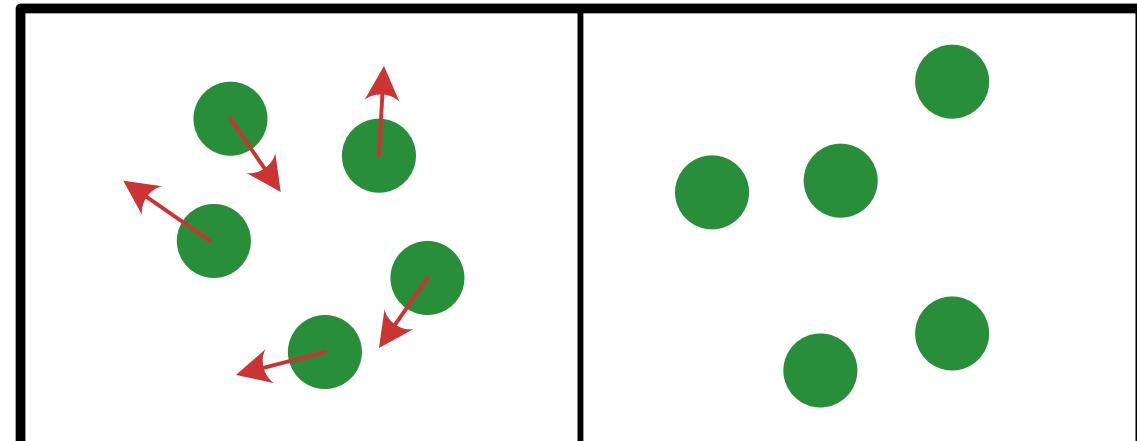


The metatomic API



metatomic

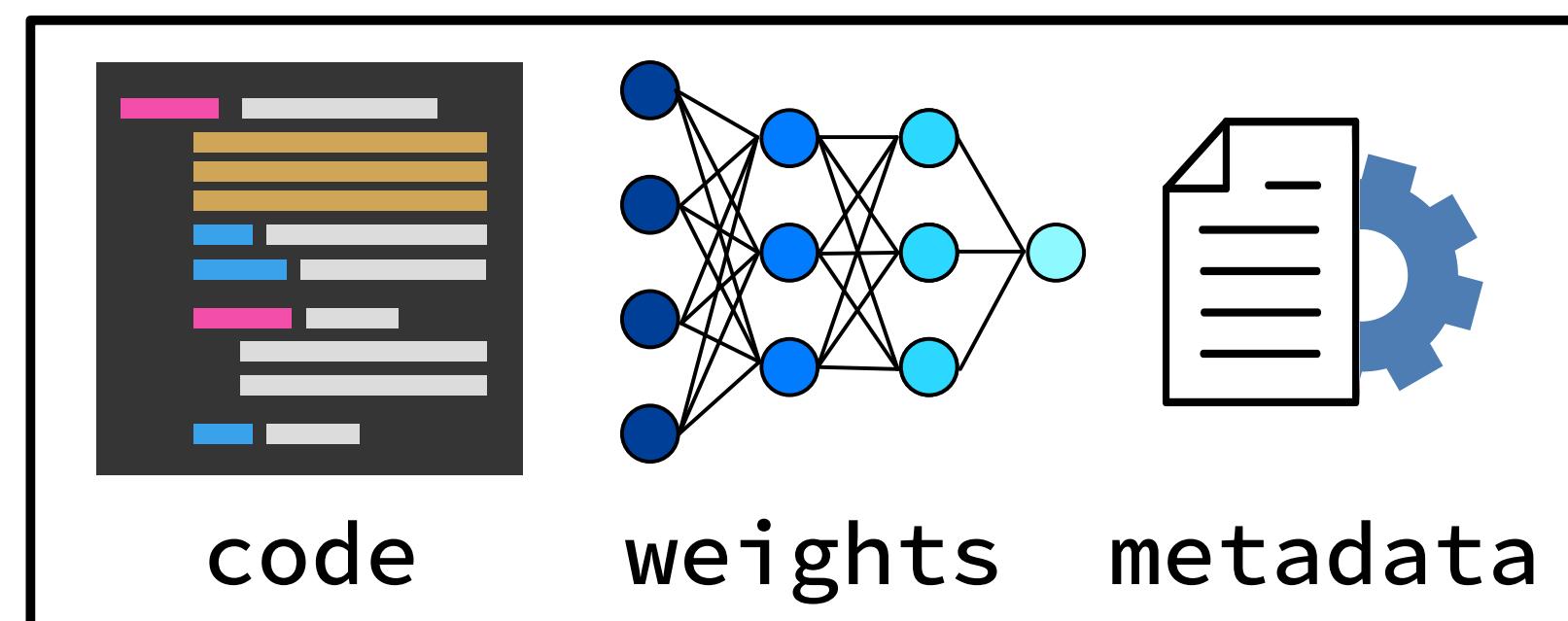
simulation engine



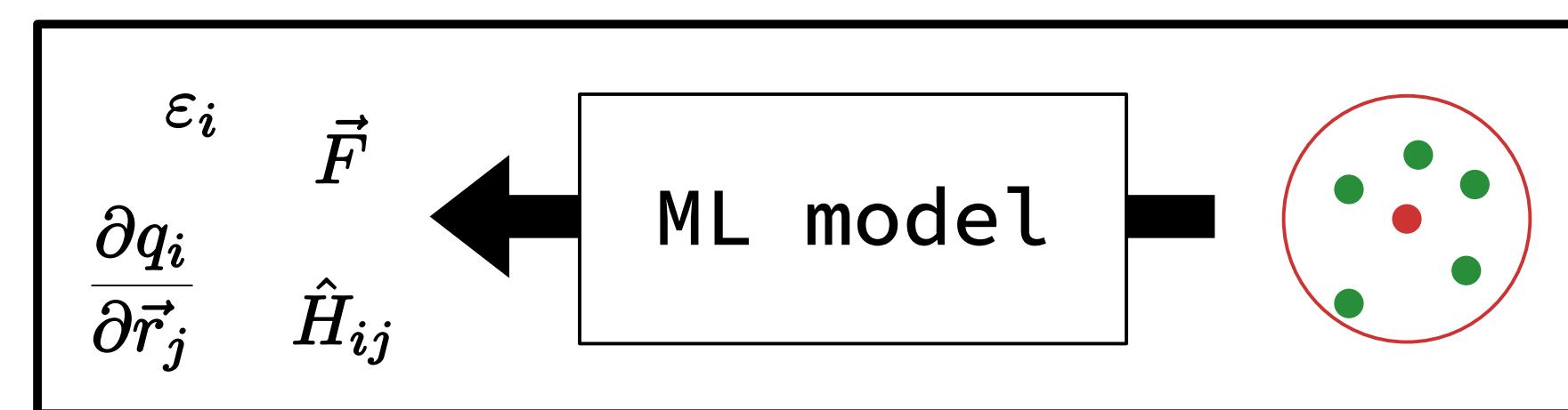
prepare



metatomic model

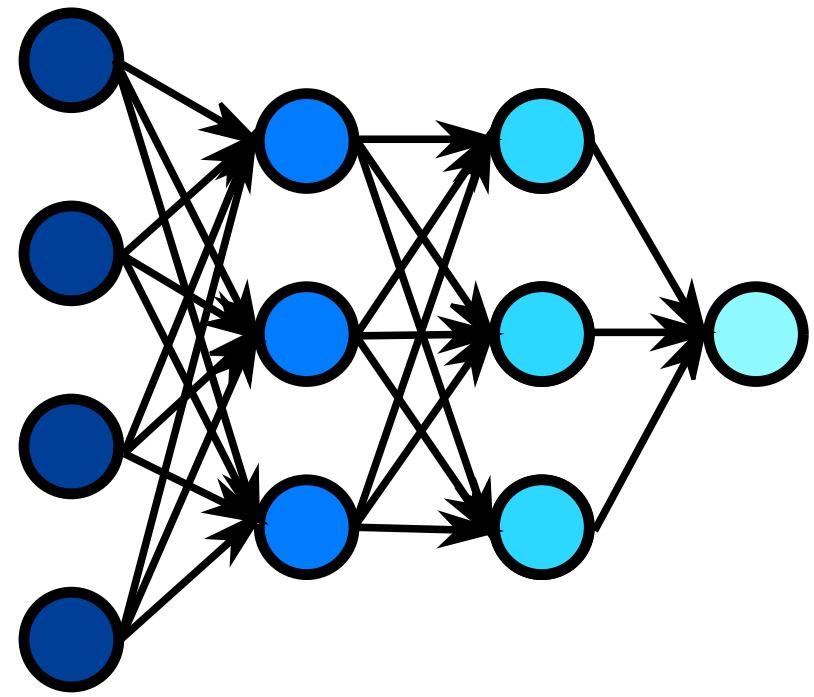


execute



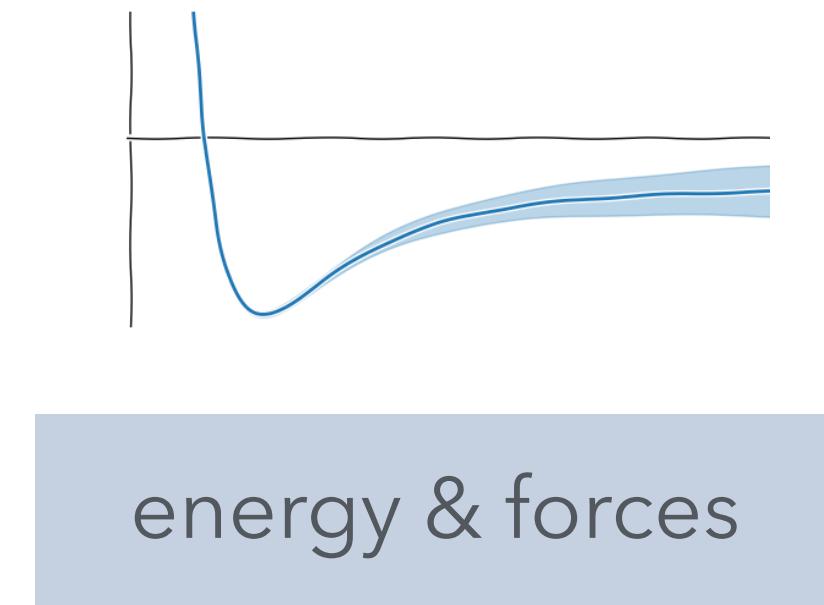
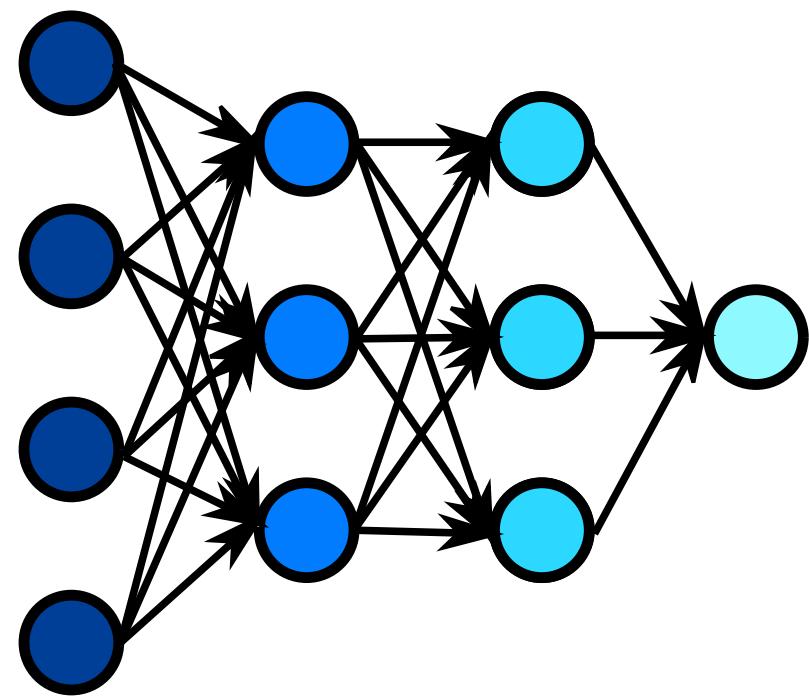
- PyTorch/TorchScript model
- Execute **any** model in **any** simulation engine, including large scale MPI tools
- Handling of units, definition of complex outputs, gradients
- Metadata and capabilities inside the model

One model, many outputs



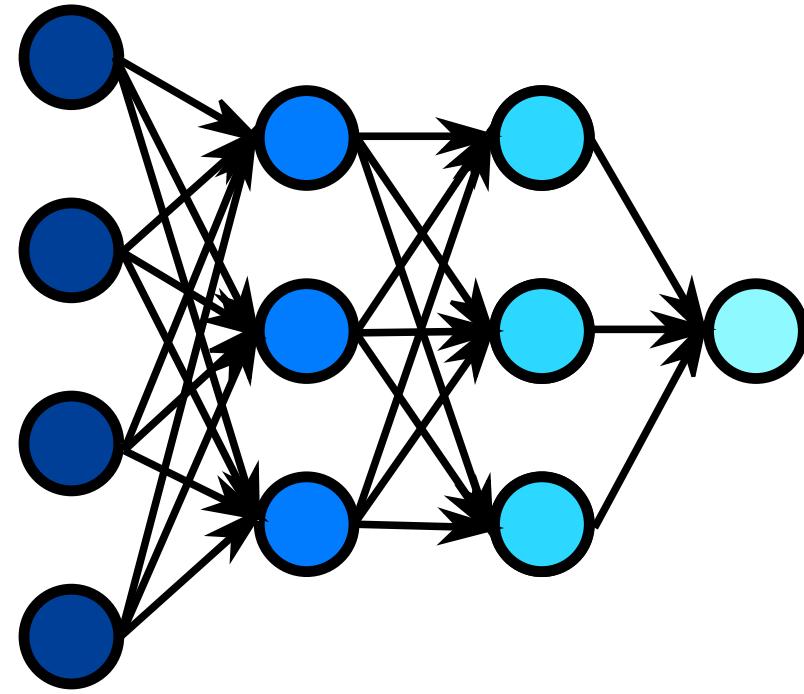
metatomic model

One model, many outputs

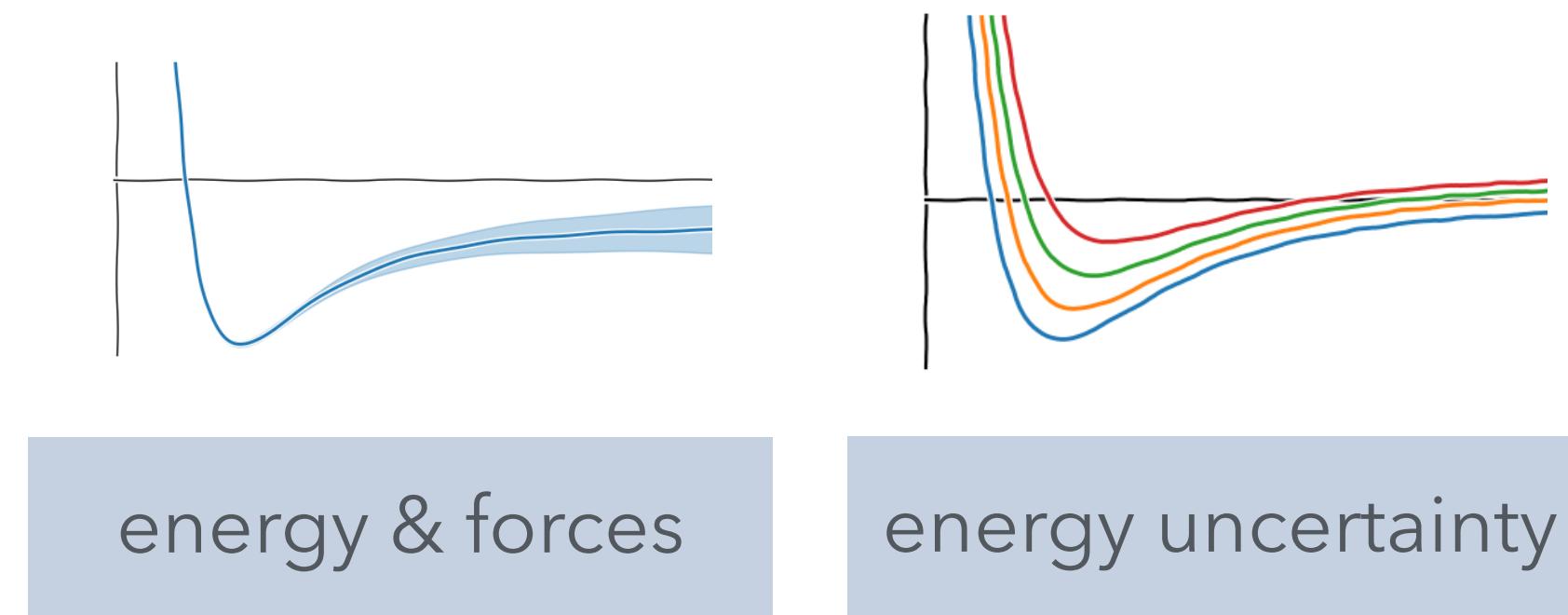


metatomic model

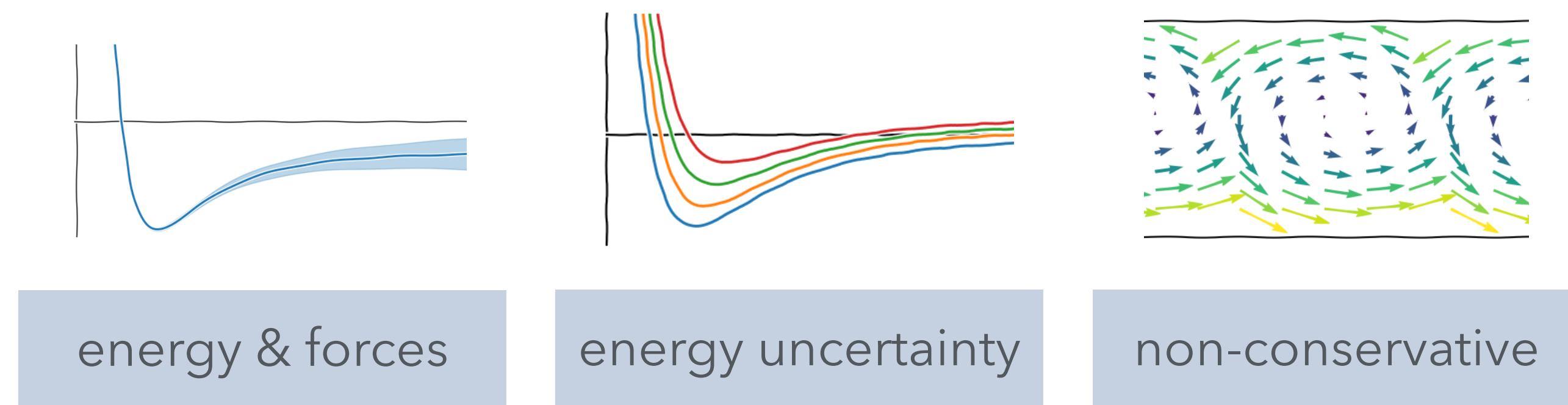
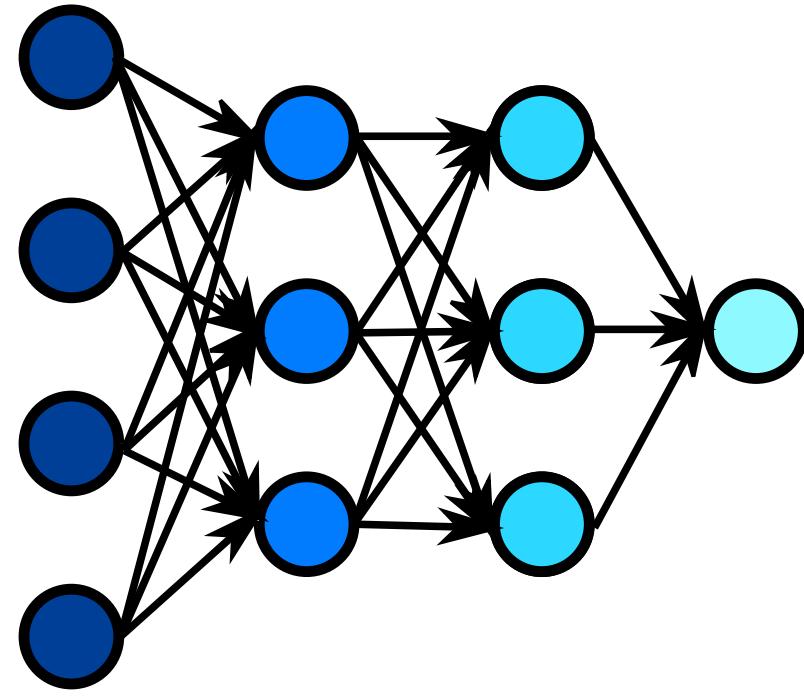
One model, many outputs



metatomic model

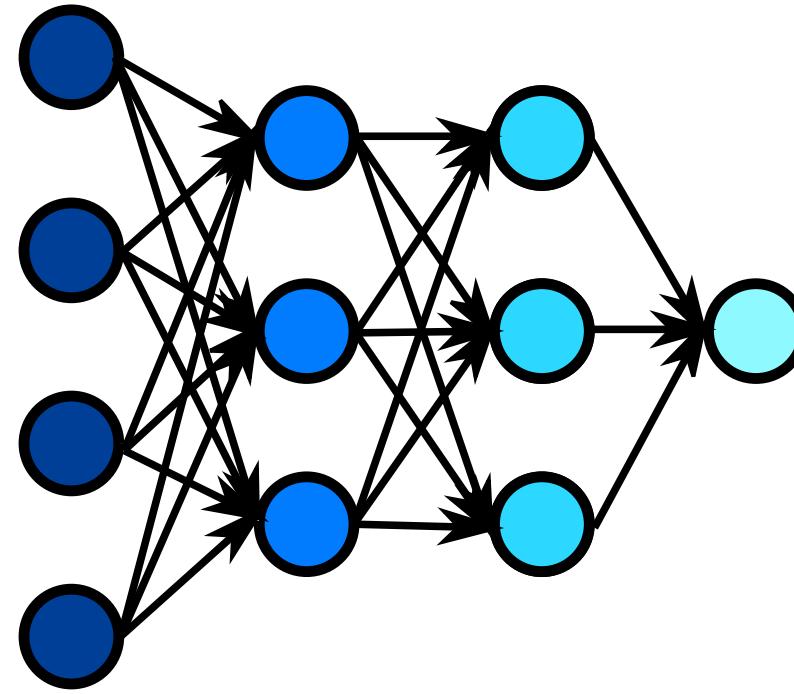


One model, many outputs

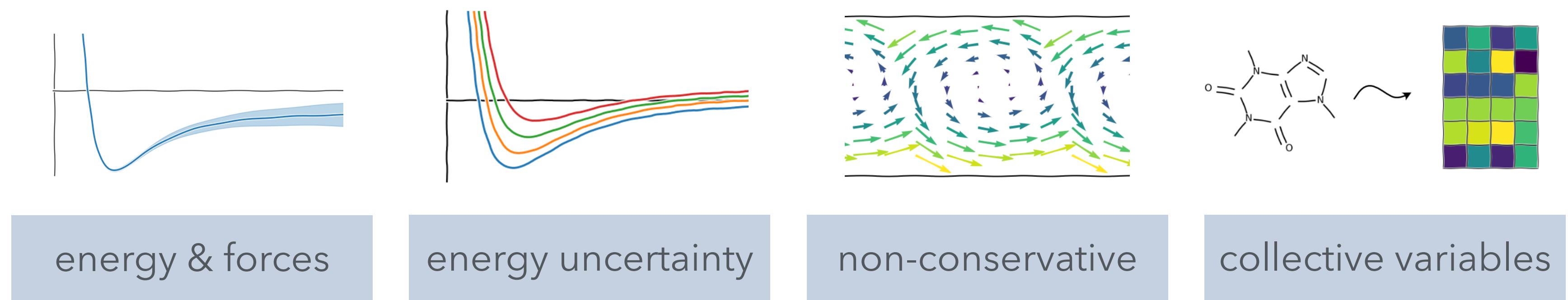


metatomic model

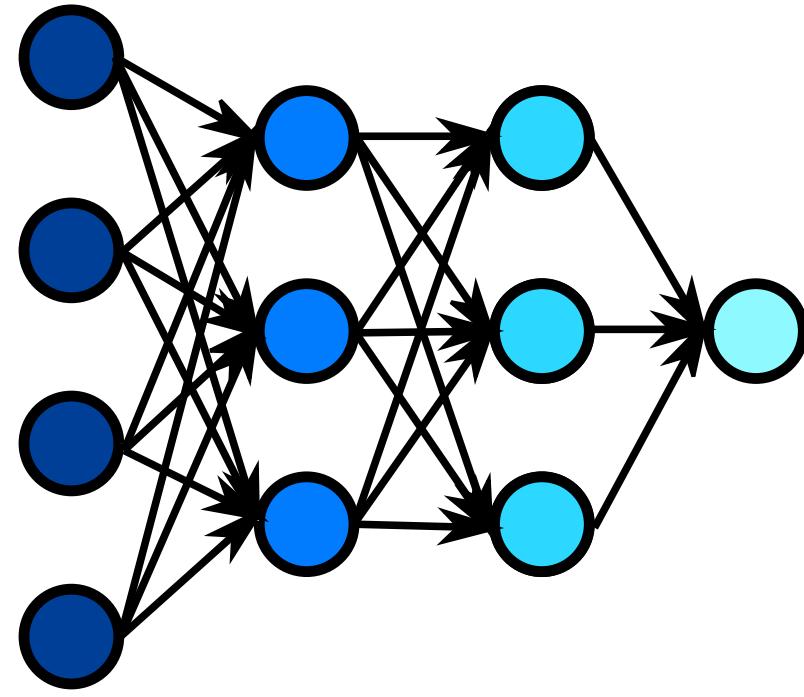
One model, many outputs



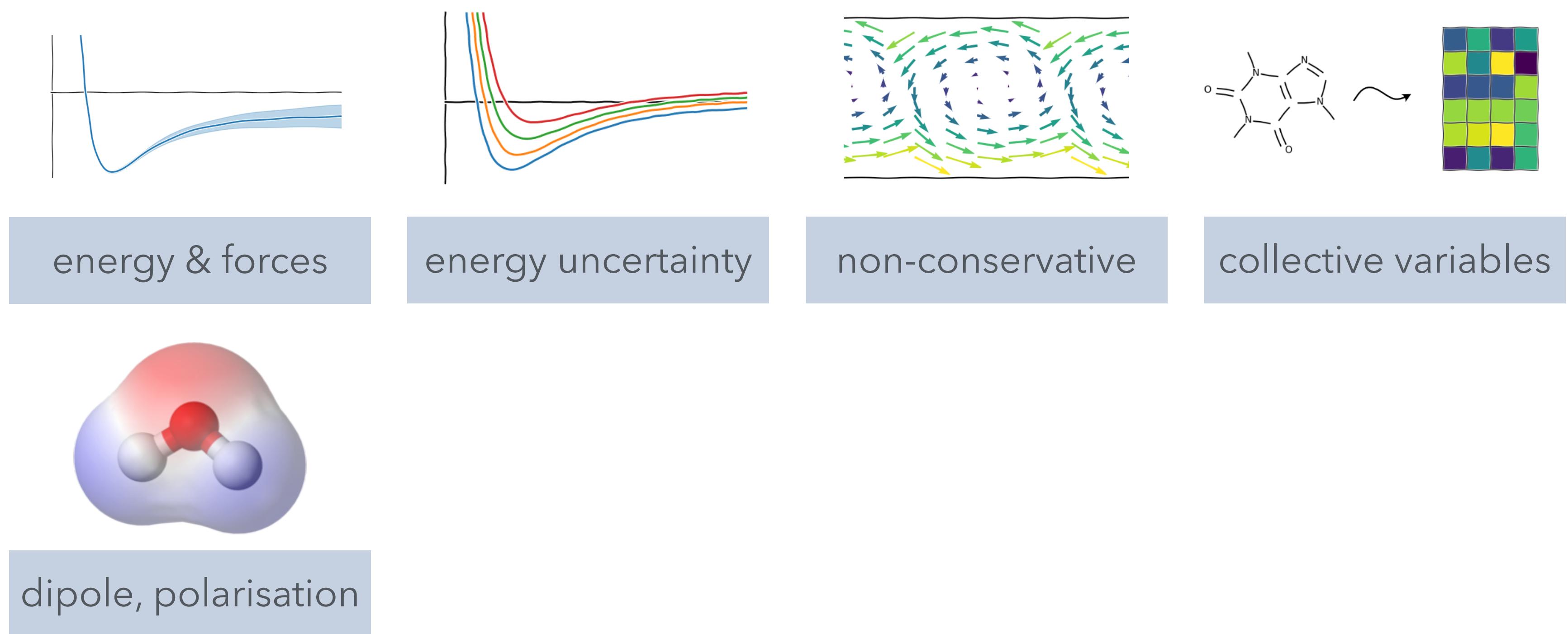
metatomic model



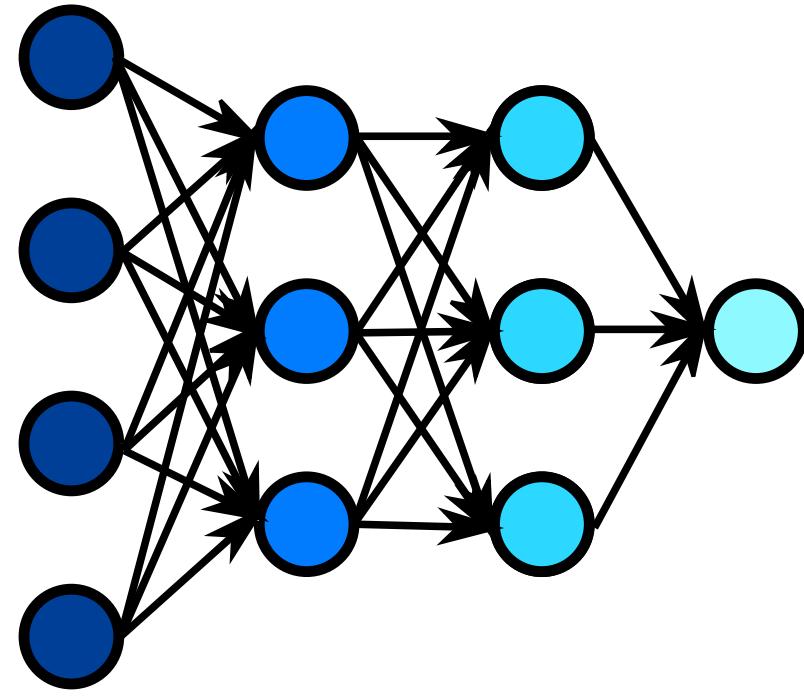
One model, many outputs



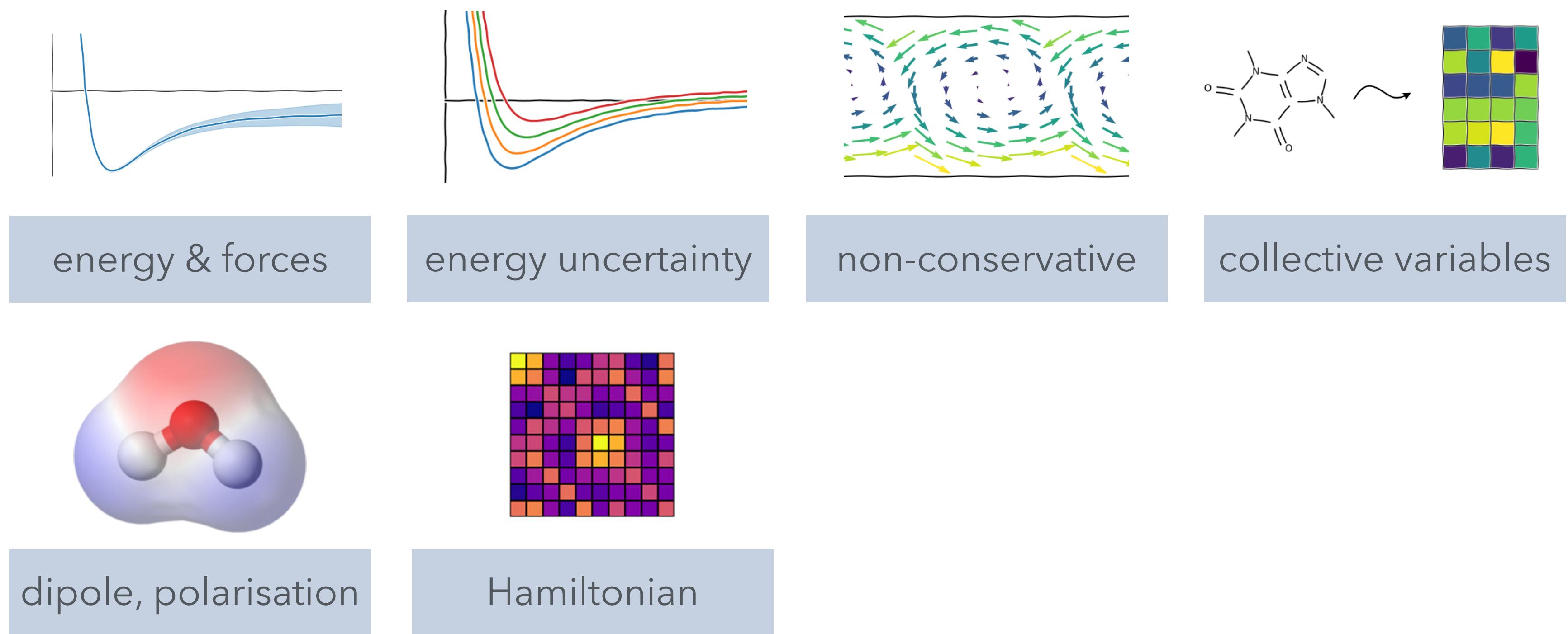
metatomic model



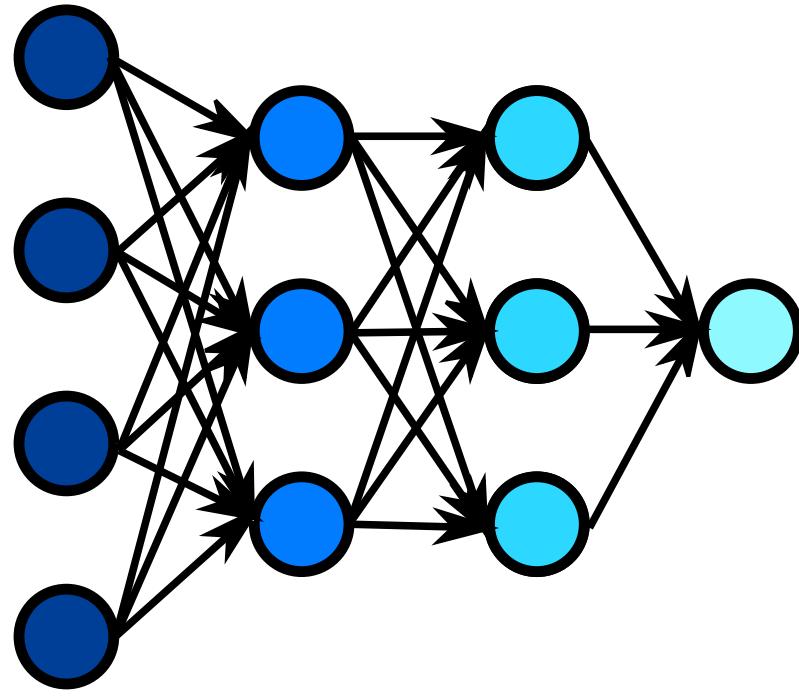
One model, many outputs



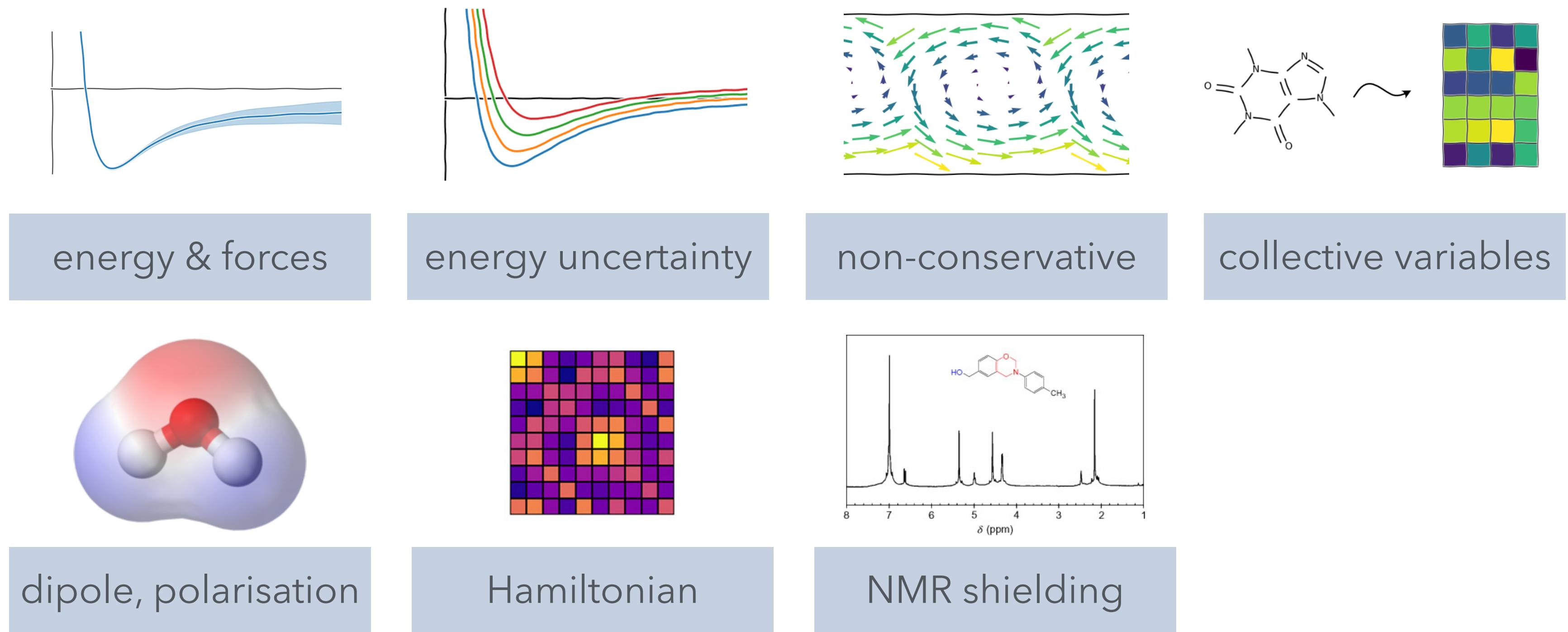
metatomic model



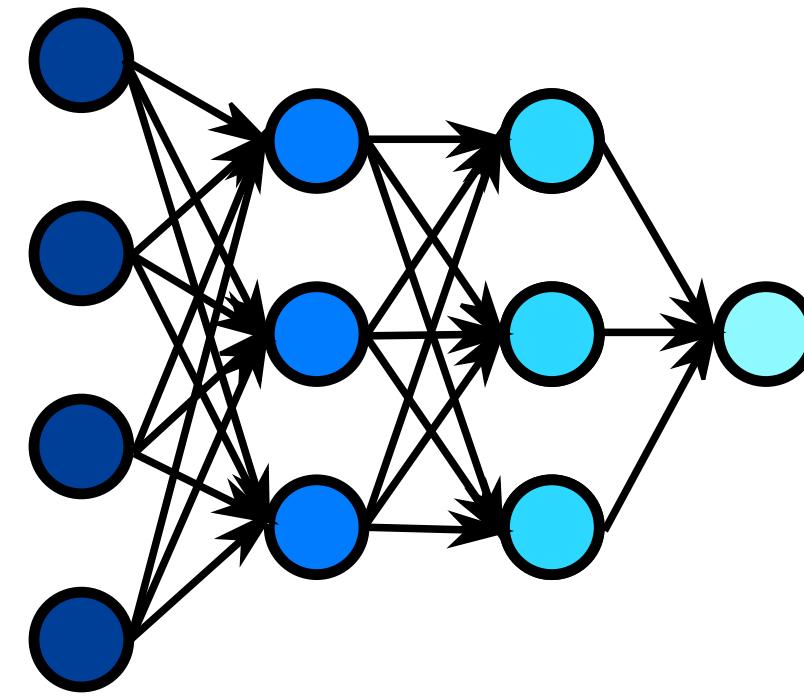
One model, many outputs



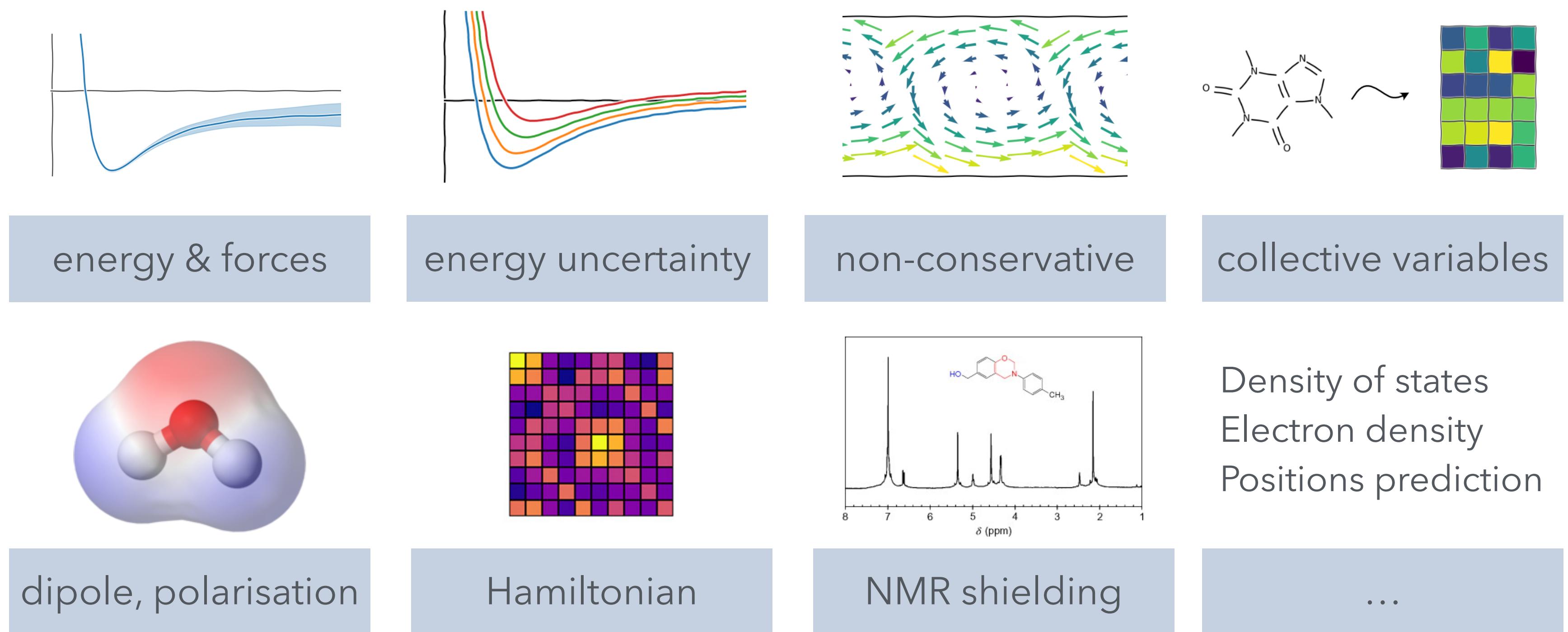
metatomic model



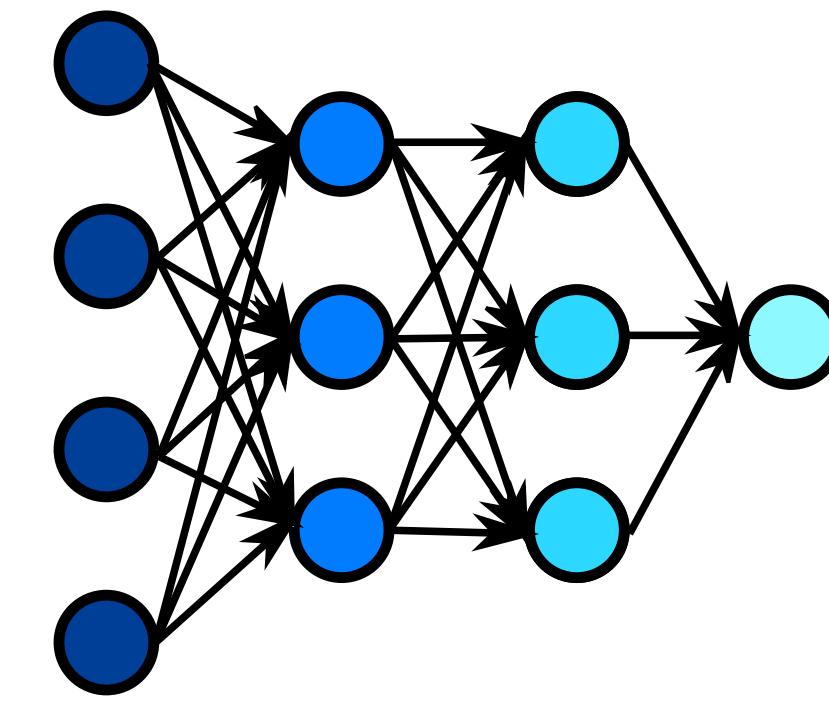
One model, many outputs



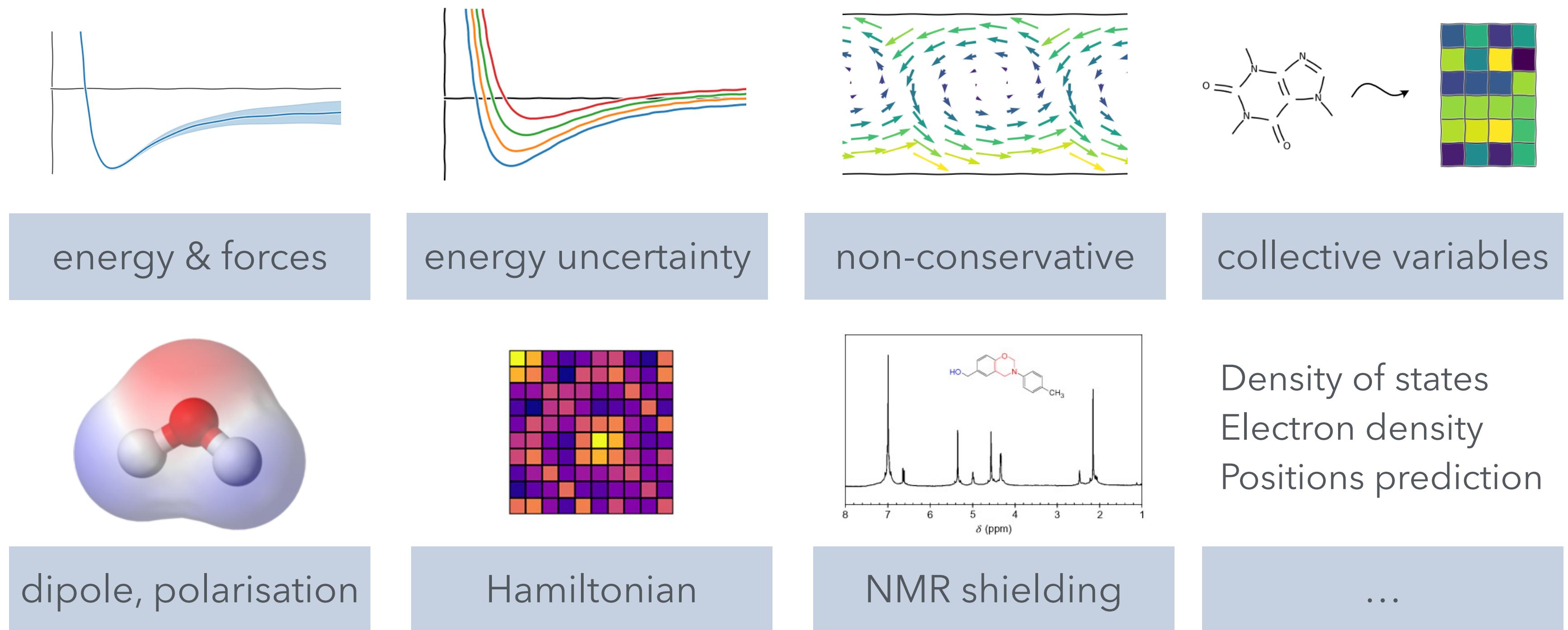
metatomic mode



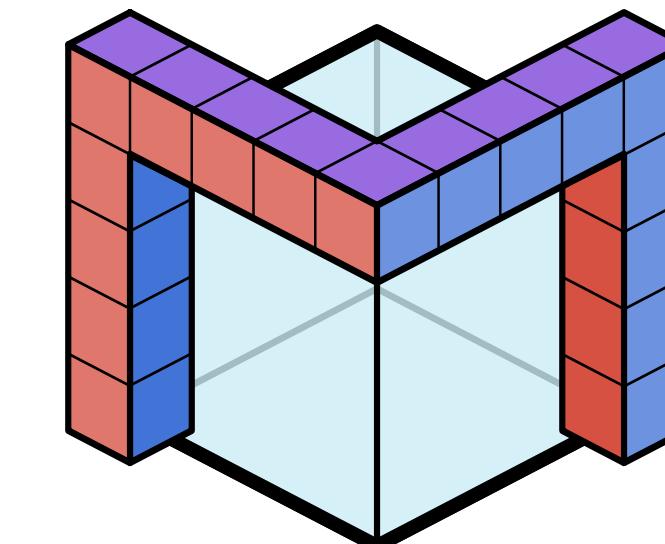
One model, many outputs



metatomic mode



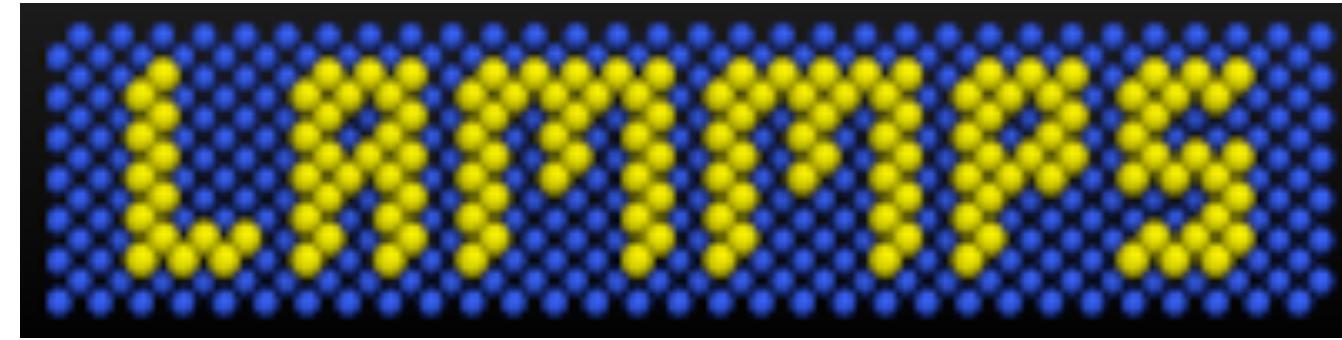
Store and exchange all kind of output data through metatensor



metatensor

<https://docs.metatensor.org/>

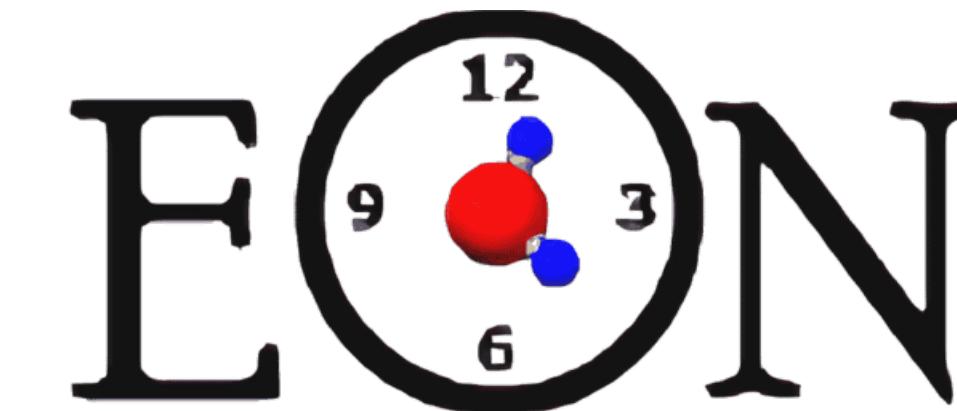
Interfaces to metatomic



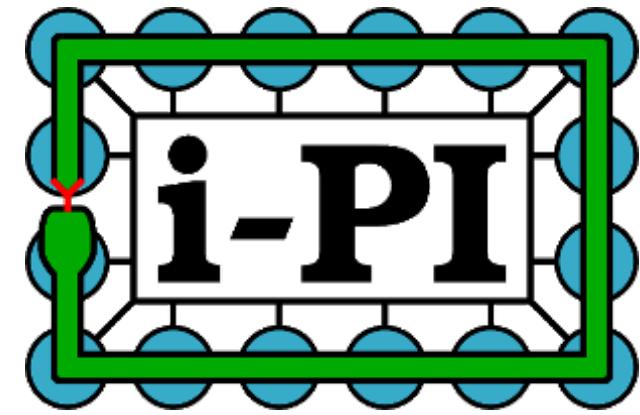
Large scale simulations
GPU acceleration with kokkos

Ease of use from Python,
can evaluate all outputs

Non-conservative forces for faster simulations,
corrected with multiple time stepping

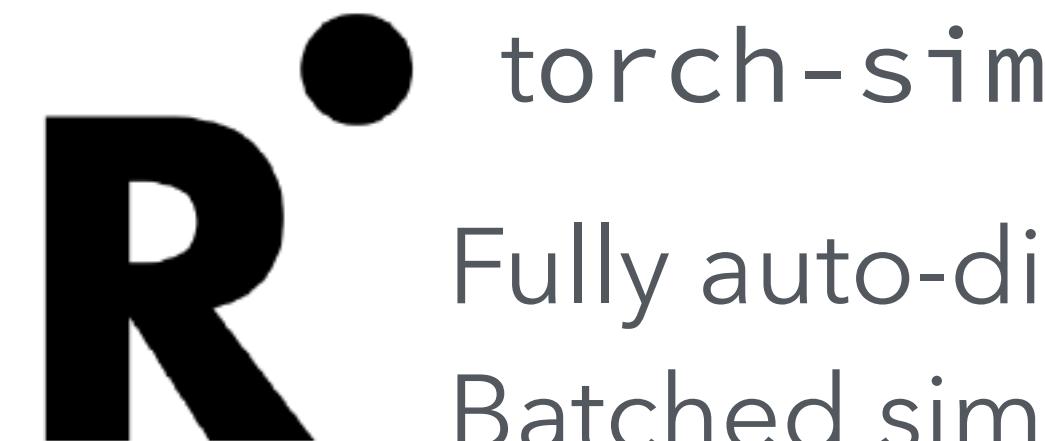


Long timescale processes
Adaptative KMC
Nudged Elastic Bands



Advanced time integrations,
including nuclear quantum effects

Propagate uncertainty from energy
to macroscopic observables



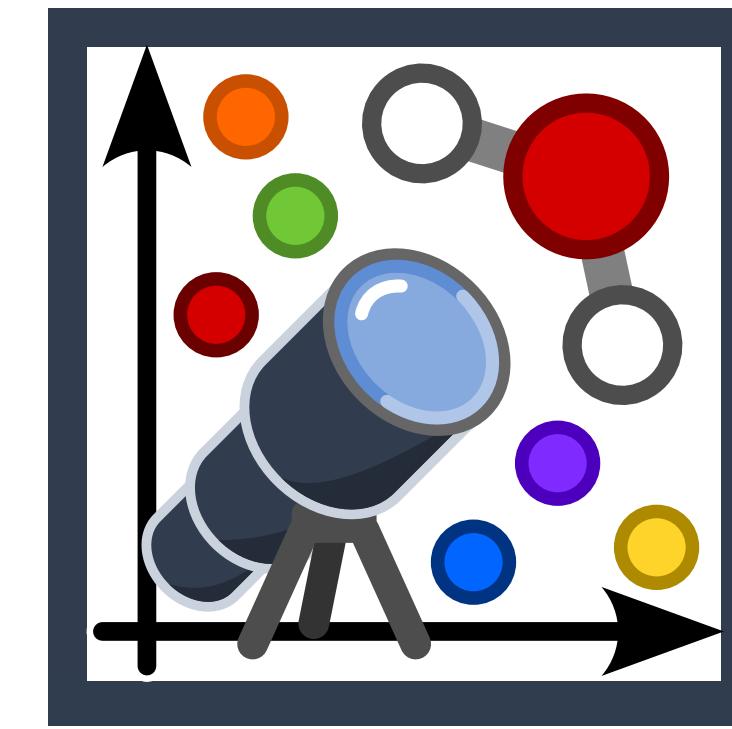
Fully auto-differentiable simulations
Batched simulations

Interfaces to metatomic



Enhanced sampling

Define custom CV with
Python code



Interactive data exploration

Interactive CV creation

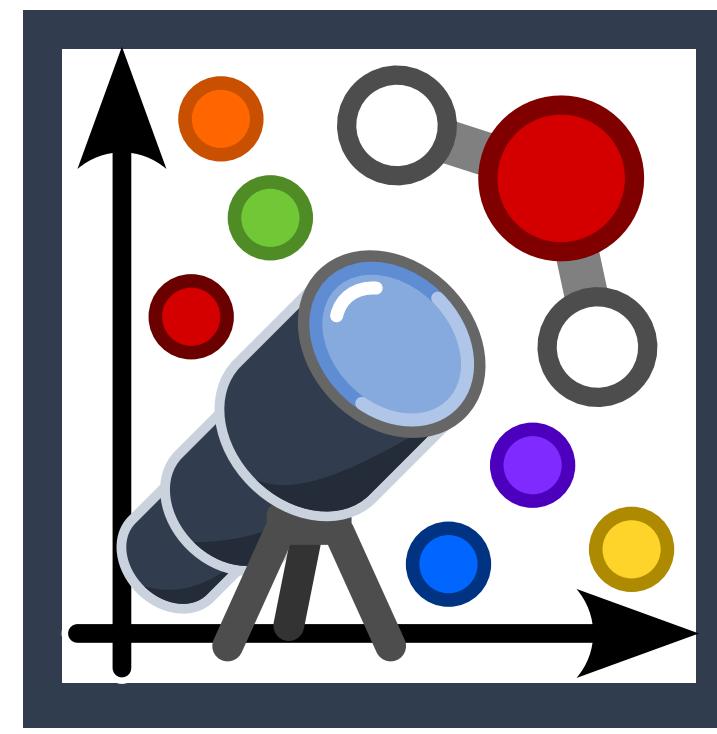
<https://chemiscope.org>

Interfaces to metatomic



Enhanced sampling

Define custom CV with
Python code



Interactive data exploration

Interactive CV creation

<https://chemiscope.org>

Published models using metatomic

PET-MAD: universal interatomic potential

B5.40

PET-MAD-DOS: universal DOS model

B5.23

ShiftML: NMR shielding prediction

C7.7

FlashMD: direct position predictions

Interfaces to metatomic



Enhanced sampling

Define custom CV with
Python code



Interactive data exploration

Interactive CV creation

<https://chemiscope.org>

Published models using metatomic

PET-MAD: universal interatomic potential

B5.40

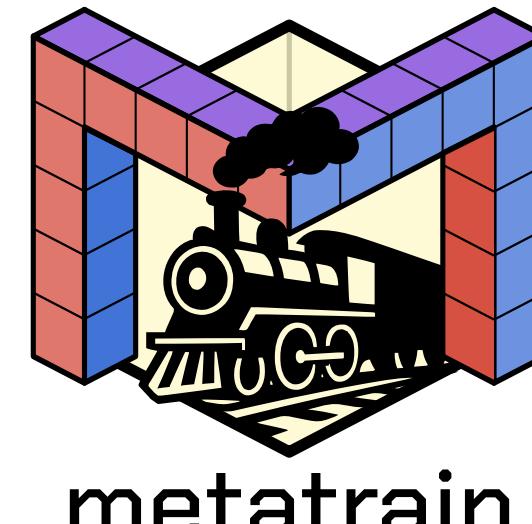
PET-MAD-DOS: universal DOS model

B5.23

ShiftML: NMR shielding prediction

C7.7

FlashMD: direct position predictions



Train your own model with metatrain!

From kernel methods to equivariant
and unconstrained deep learning

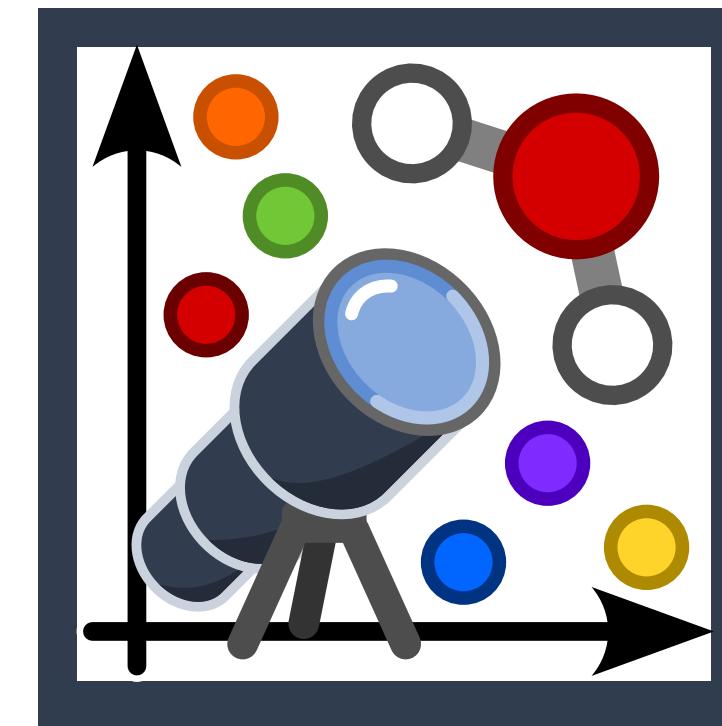
<https://github.com/metatensor/metatrain>

Interfaces to metatomic



Enhanced sampling

Define custom CV with
Python code



Interactive data exploration

Interactive CV creation

<https://chemiscope.org>

Published models using metatomic

PET-MAD: universal interatomic potential

B5.40

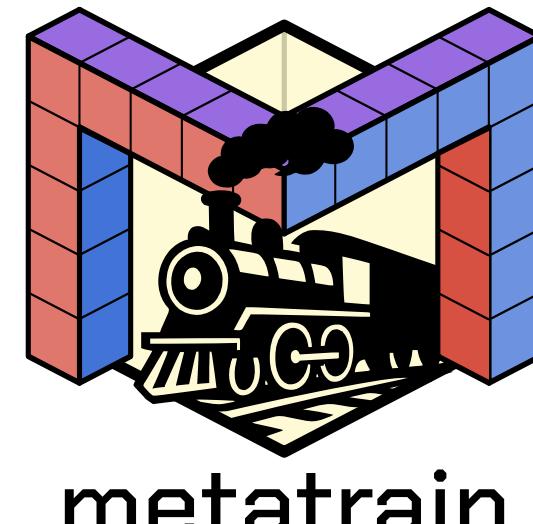
PET-MAD-DOS: universal DOS model

B5.23

ShiftML: NMR shielding prediction

C7.7

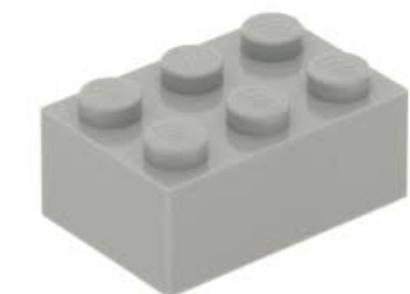
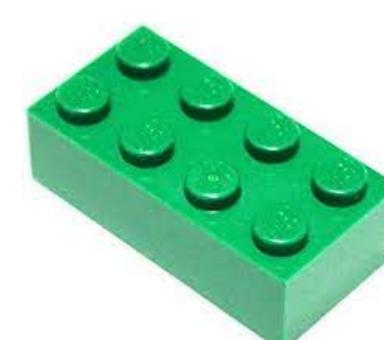
FlashMD: direct position predictions



Train your own model with metatrain!

From kernel methods to equivariant
and unconstrained deep learning

<https://github.com/metatensor/metatrain>



Assemble custom model

Applications

PET-MAD



Talk B5.40, Tue 11am – Talk B5.23, Thu noon
Poster C1.16 – Poster C1.22 – Poster C4.43

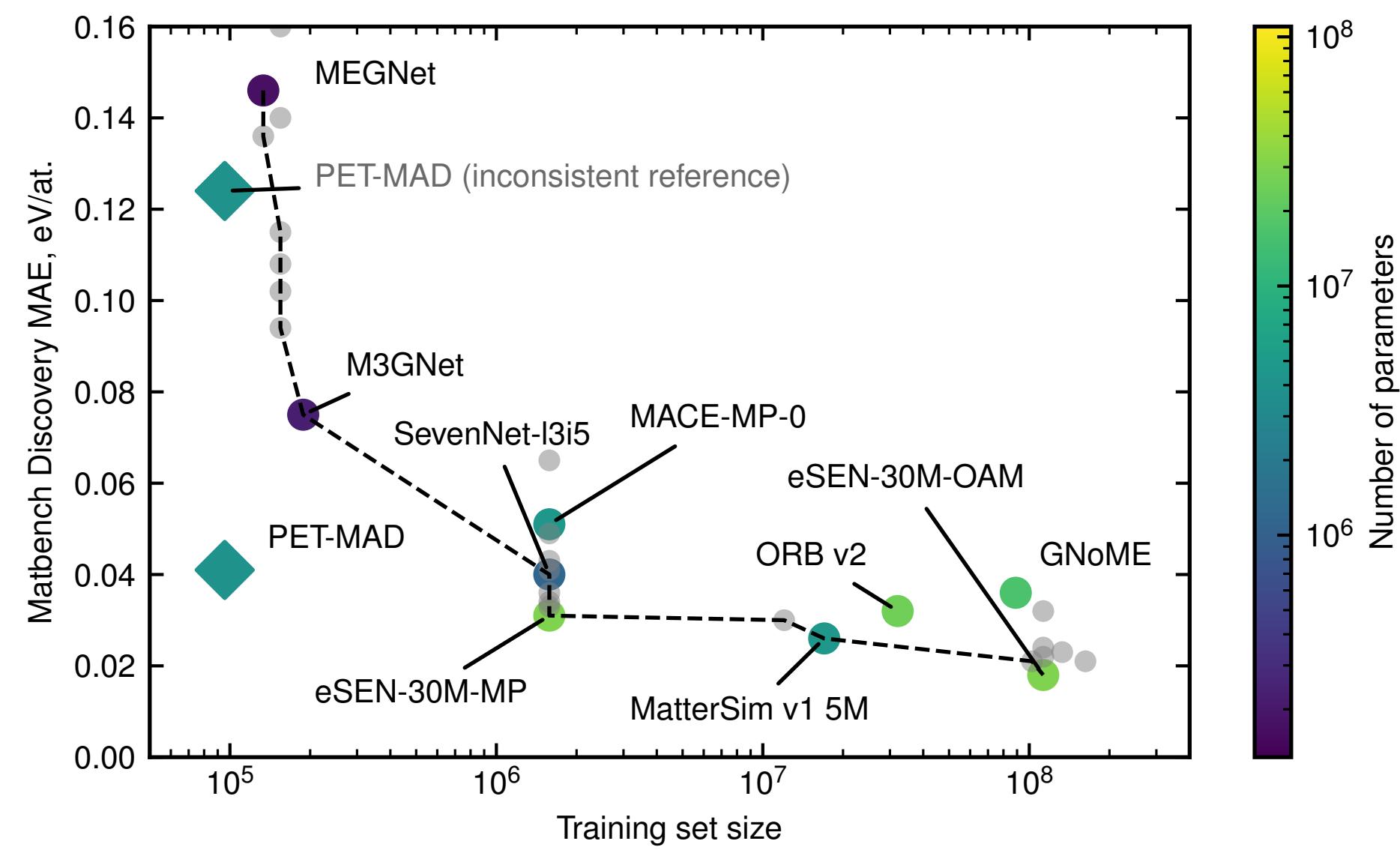
Applications



Talk B5.40, Tue 11am – Talk B5.23, Thu noon
Poster C1.16 – Poster C1.22 – Poster C4.43

PET-MAD

- Point Edge Transformer model with Massive Atomic Diversity dataset



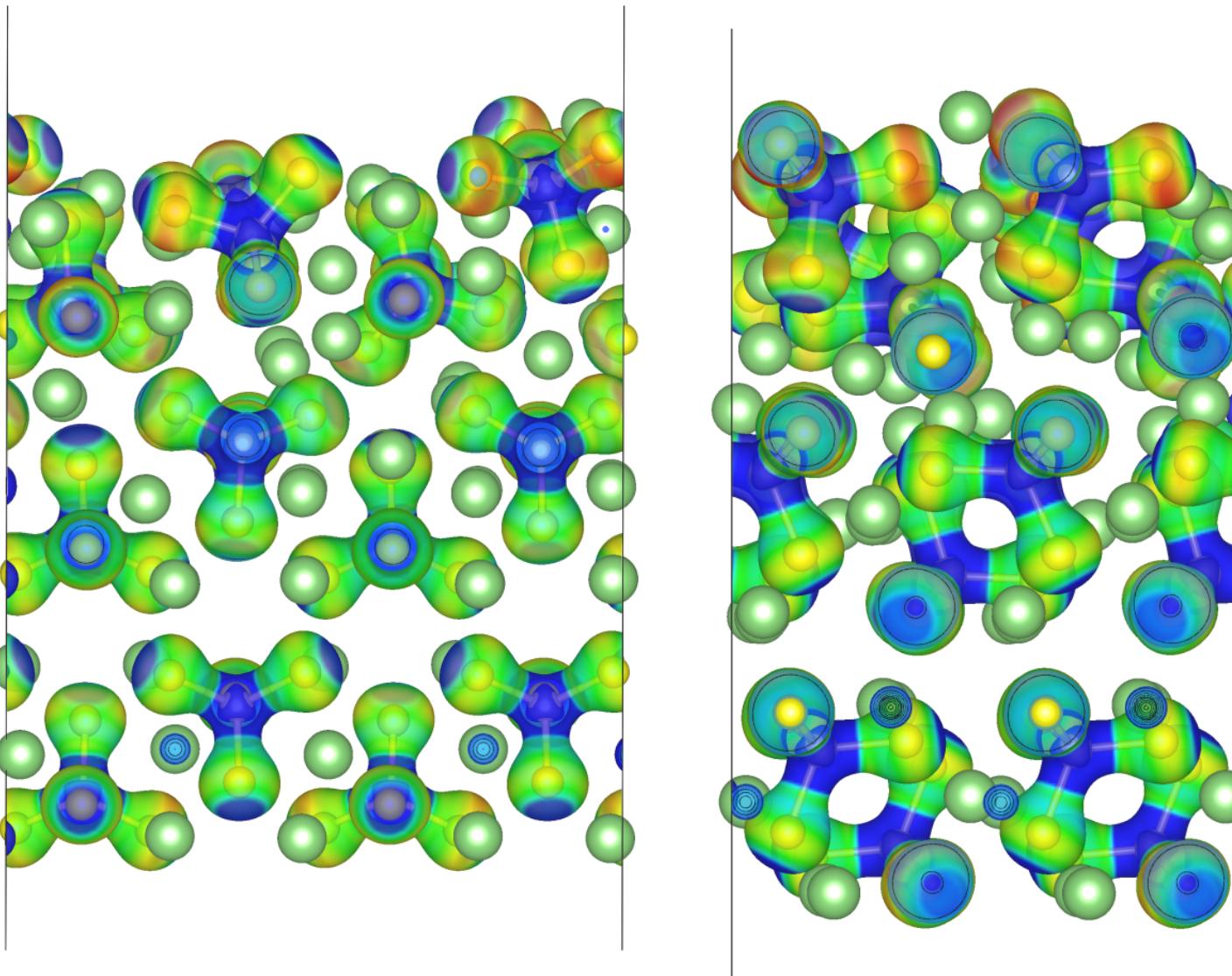
Applications



Talk B5.40, Tue 11am – Talk B5.23, Thu noon
Poster C1.16 – Poster C1.22 – Poster C4.43

PET-MAD

- Point Edge Transformer model with Massive Atomic Diversity dataset
- Example: surface reconstruction in LiPS



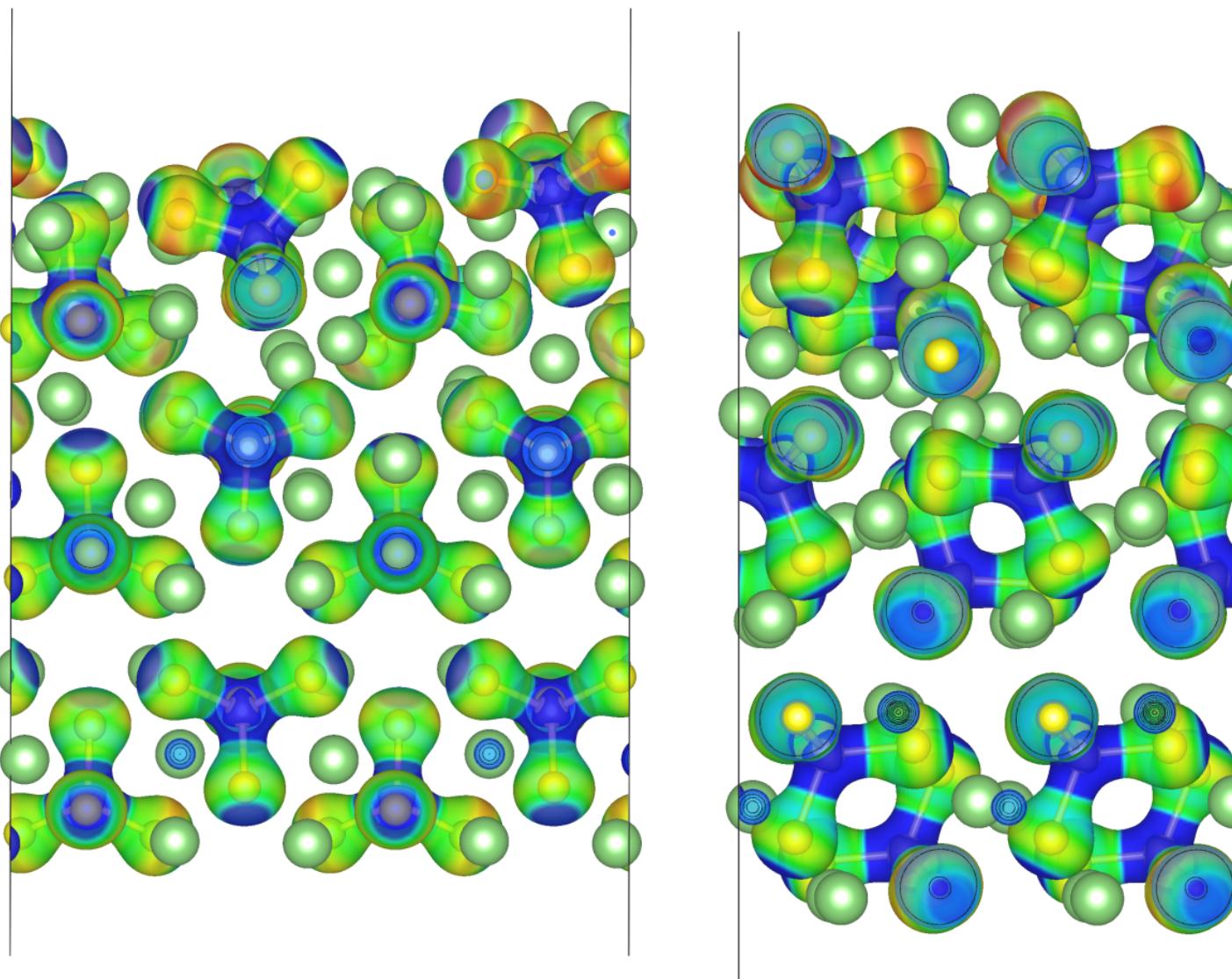
Applications



Talk B5.40, Tue 11am – Talk B5.23, Thu noon
Poster C1.16 – Poster C1.22 – Poster C4.43

PET-MAD

- Point Edge Transformer model with Massive Atomic Diversity dataset
- Example: surface reconstruction in LiPS
- Simulations in LAMMPS & i-PI, using PET CV in chemiscope and PLUMED



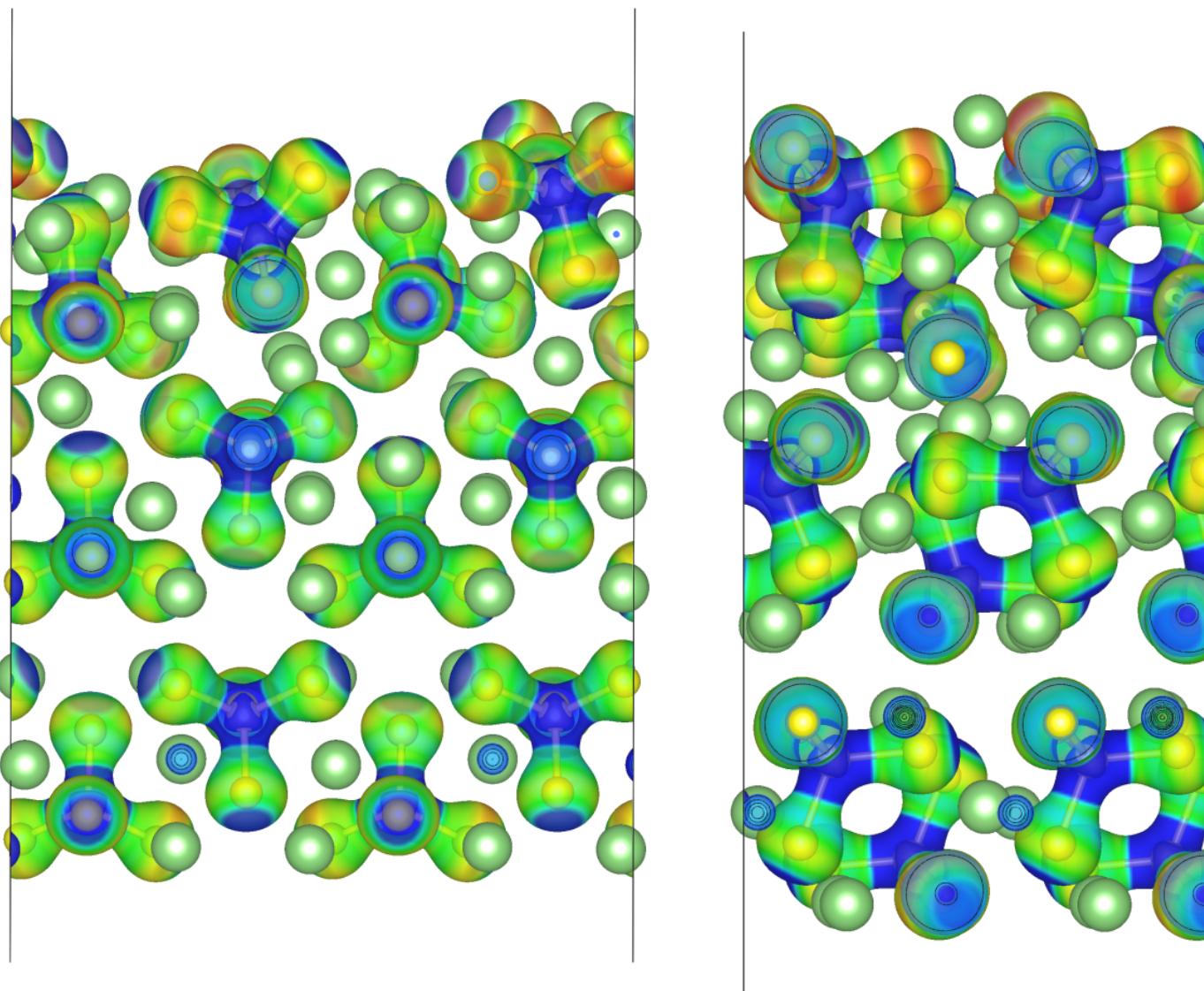
Applications



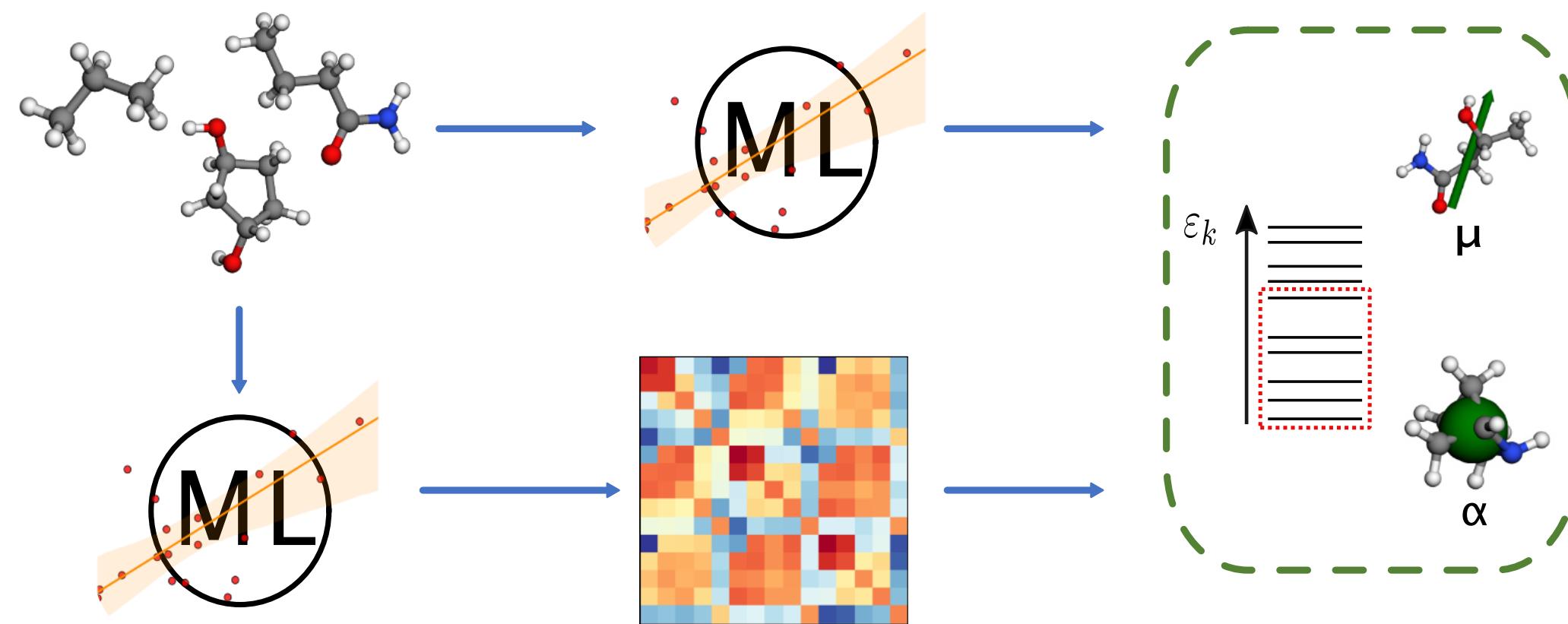
Talk B5.40, Tue 11am – Talk B5.23, Thu noon
Poster C1.16 – Poster C1.22 – Poster C4.43
Poster B5.59

PET-MAD

- Point Edge Transformer model with Massive Atomic Diversity dataset
- Example: surface reconstruction in LiPS
- Simulations in LAMMPS & i-PI, using PET CV in chemiscope and PLUMED



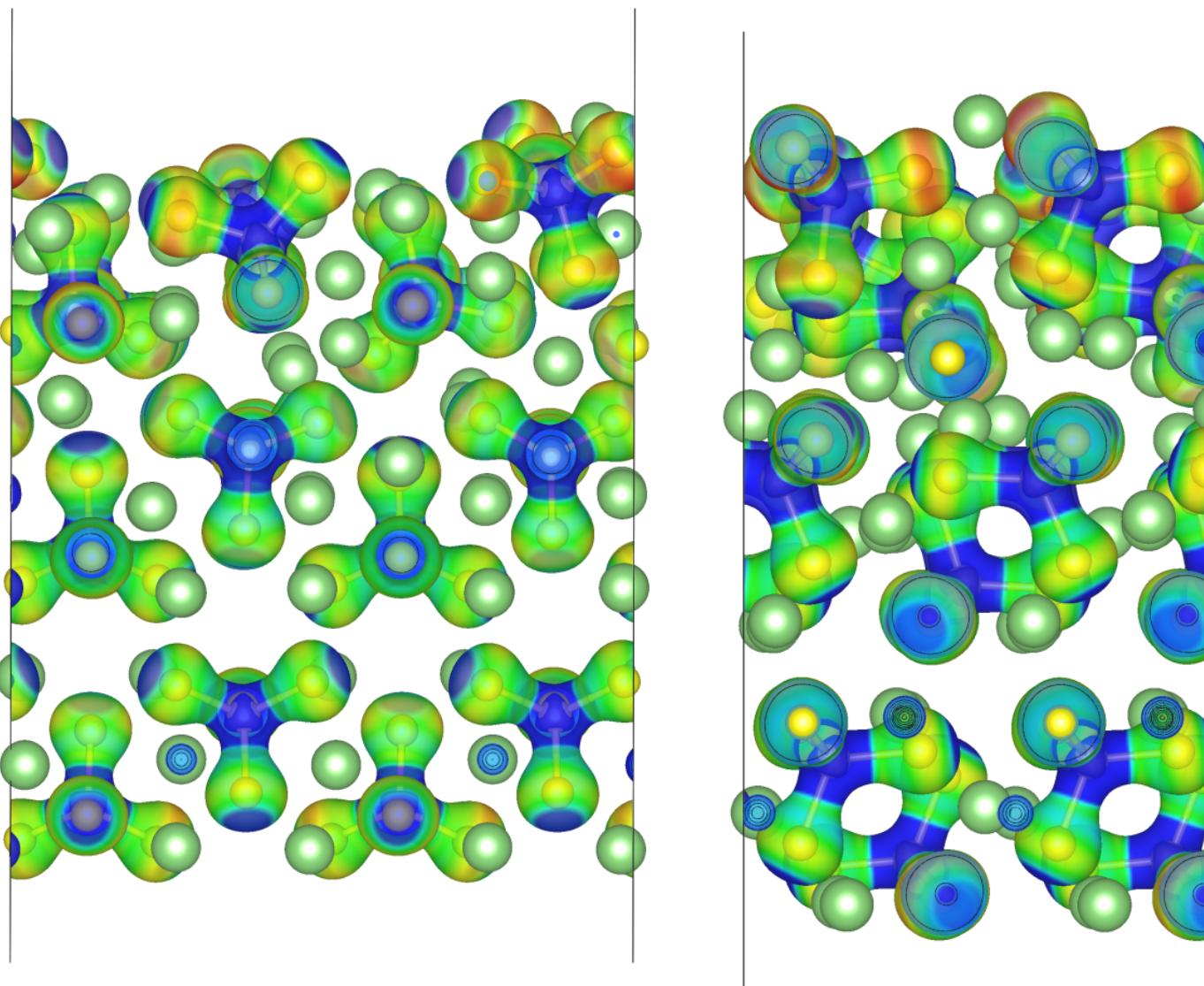
Hamiltonian



Applications

PET-MAD

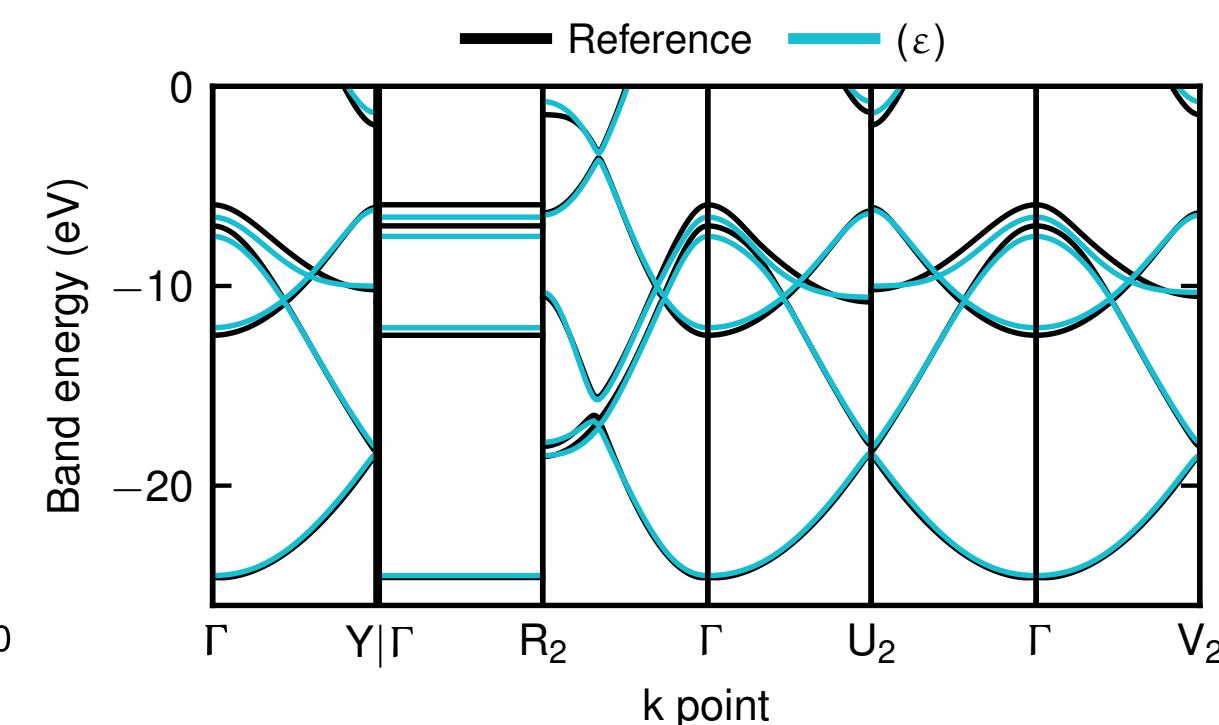
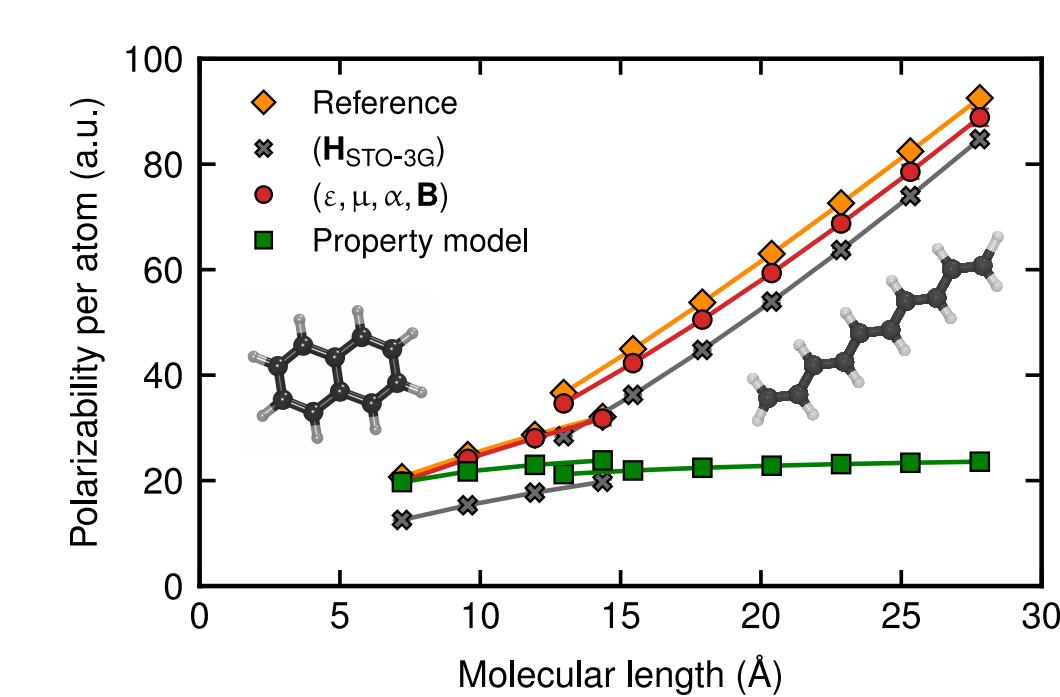
- Point Edge Transformer model with Massive Atomic Diversity dataset
- Example: surface reconstruction in LiPS
- Simulations in LAMMPS & i-PI, using PET CV in chemiscope and PLUMED



Hamiltonian



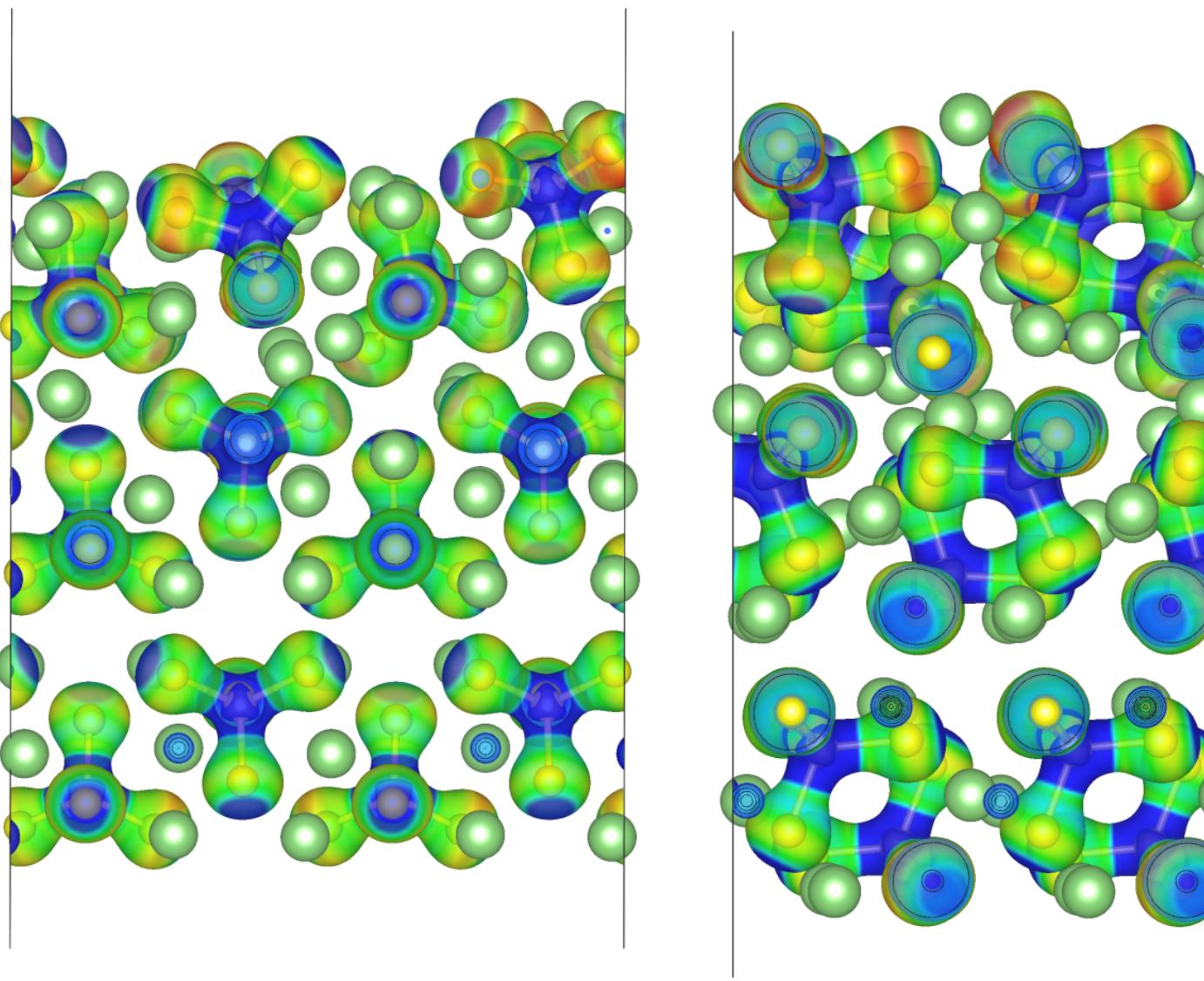
Talk B5.40, Tue 11am – Talk B5.23, Thu noon
Poster C1.16 – Poster C1.22 – Poster C4.43
Poster B5.59



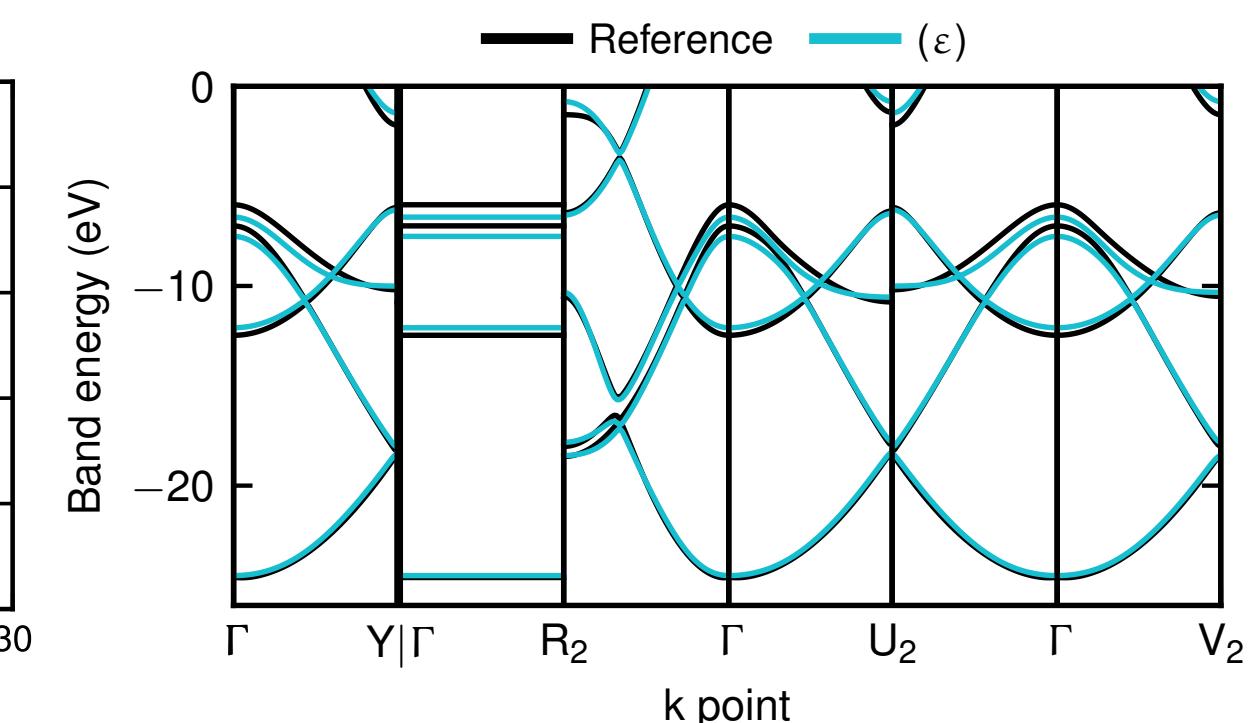
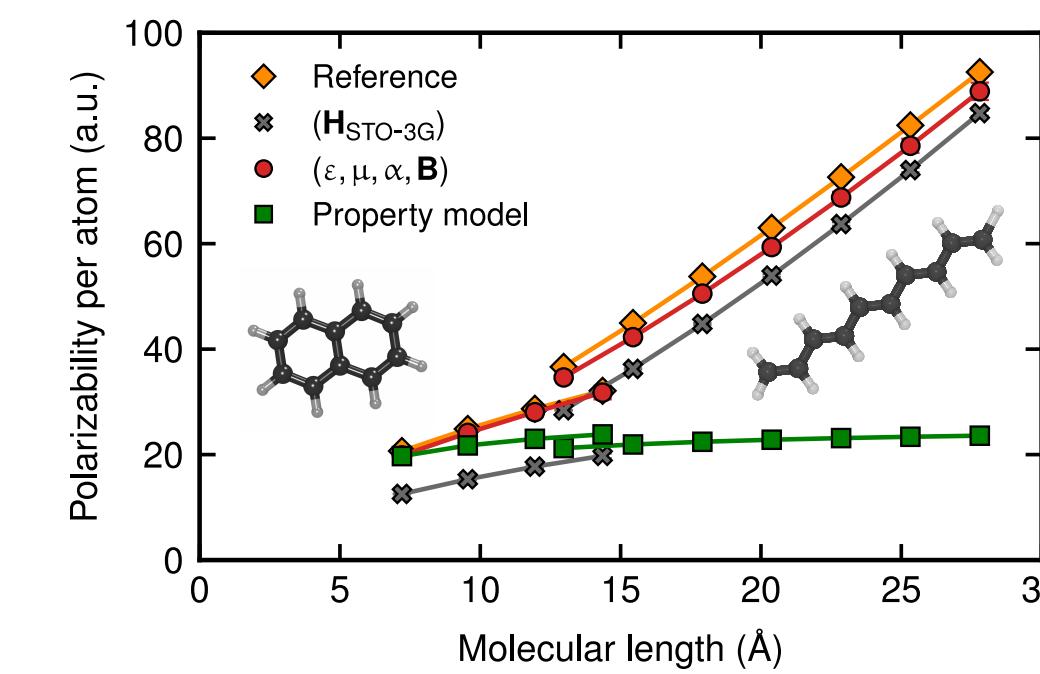
Applications

PET-MAD

- Point Edge Transformer model with Massive Atomic Diversity dataset
- Example: surface reconstruction in LiPS
- Simulations in LAMMPS & i-PI, using PET CV in chemiscope and PLUMED

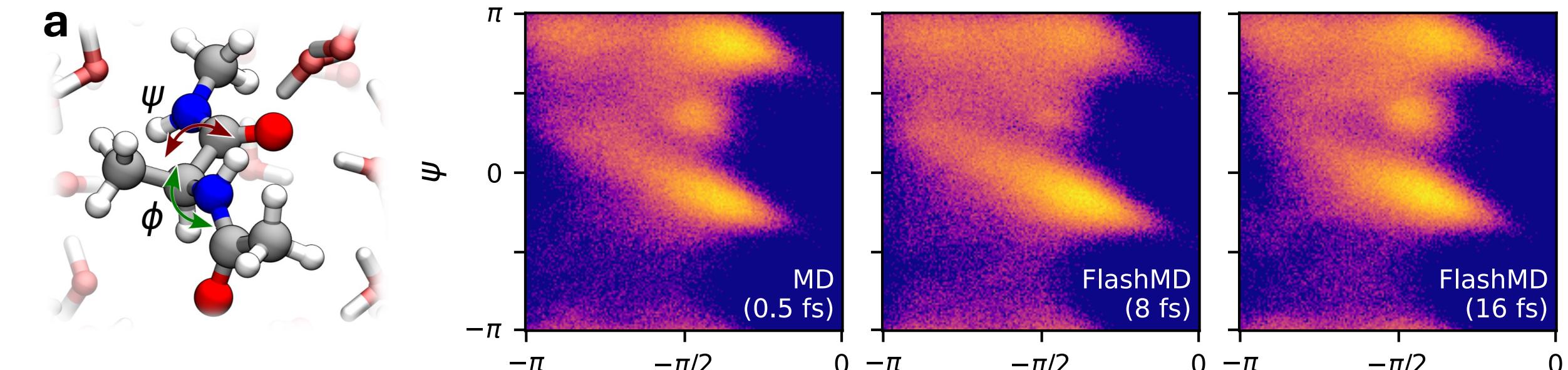


Hamiltonian



FlashMD

- Bypass time integration, directly predict positions and momenta after 8 fs / 16 fs / 32 fs



Talk B5.40, Tue 11am – Talk B5.23, Thu noon
Poster C1.16 – Poster C1.22 – Poster C4.43
Poster B5.59

Try it!

Run simulations with PET-MAD

```
conda install -c metatensor lammps-metatomic  
https://github.com/lab-cosmo/pet-mad
```

Try it!

Run simulations with PET-MAD

```
conda install -c metatensor lammps-metatomic
```

```
https://github.com/lab-cosmo/pet-mad
```

Make your model compatible

Use it in your simulation code

```
https://docs.metatensor.org/metatomic
```

Try it!

Run simulations with PET-MAD

```
conda install -c metatensor lammps-metatomic  
https://github.com/lab-cosmo/pet-mad
```

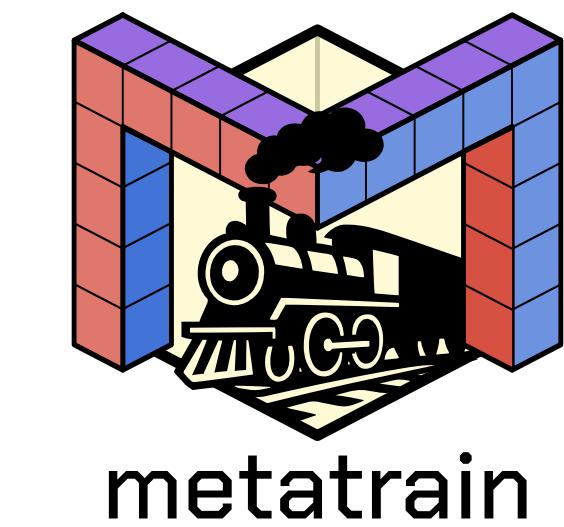
Make your model compatible

Use it in your simulation code

<https://docs.metatensor.org/metatomic>

Train your own model

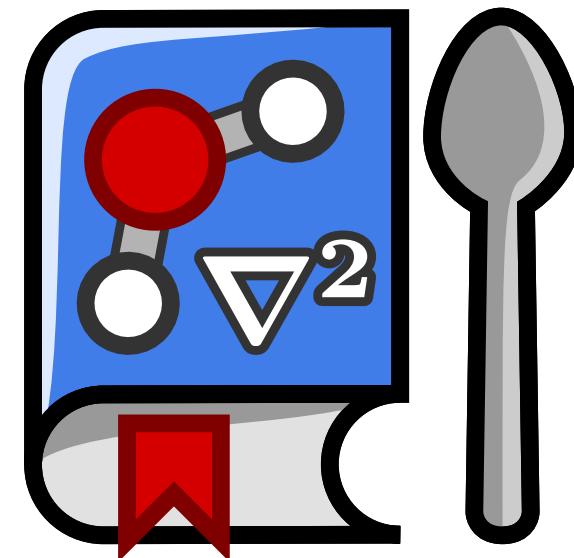
<https://github.com/metatensor/metatrain>



Try it!

Run simulations with PET-MAD

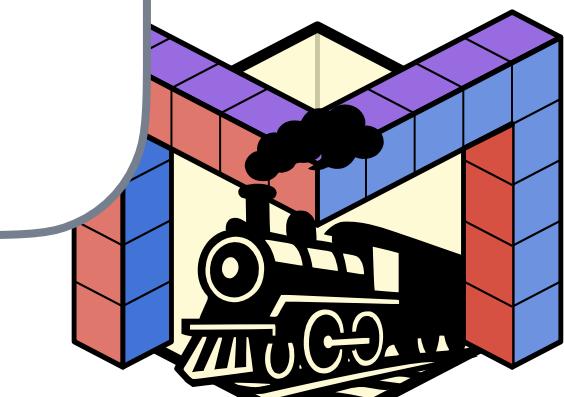
```
conda install -c metatensor lammps-metatomic
```



<https://atomistic-cookbook.org/>

Make your model compatible

Train your own model



metatrain

Use it in your simulation code

<https://github.com/metatensor/metatrain>

<https://docs.metatensor.org/metatomic>

Thanks everyone



Download these slides



Check the preprint



Platform for Advanced Scientific Computing

