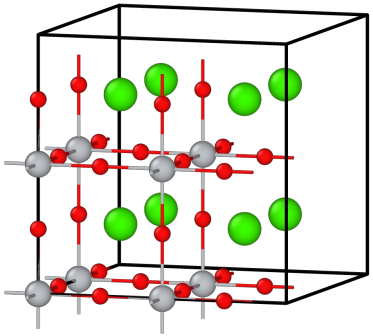
PET-MAD:
A universal interatomic potential for advanced materials modeling

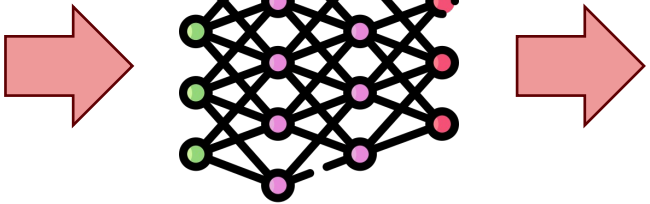
Arslan Mazitov
EPFL, Lausanne, Switzerland

Metatensor Workshop
June 13, 2025

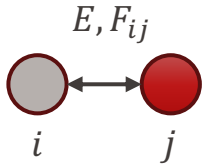
Atomistic Machine Learning



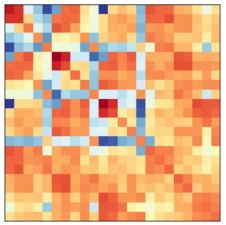
- Positions
- Cell vectors
- Chemical species



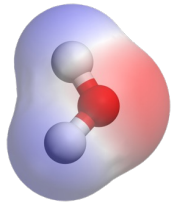
ML model



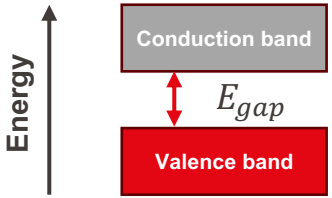
MLIP: energies, forces, stresses



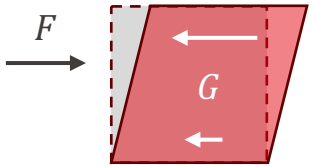
Hamiltonian eigenvalues



Charge densities

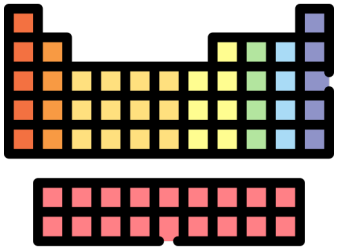


Electronic properties

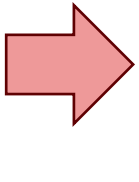


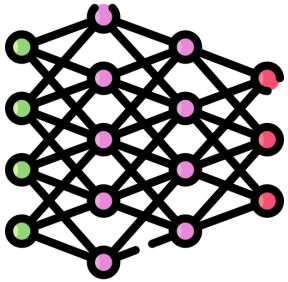
Mechanical properties

Universal MLIPs

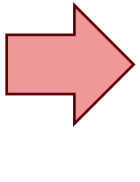


Large Materials Database





ML model



UMLIP

Works for arbitrary systems

MLIP

Works for a specific system

MACE-MP-0

MatterSim

Examples of the universal MLIPs

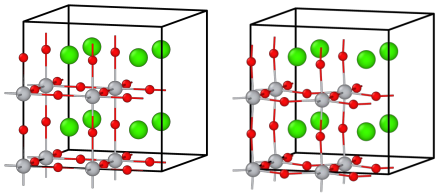
Are UMLIPs really universal?

UMLIP

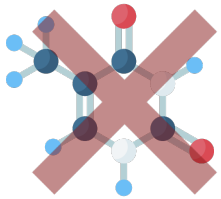
Works for arbitrary systems

… maybe
… maybe not

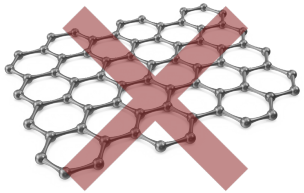
Limited structural diversity



Relaxed / non-relaxed crystals

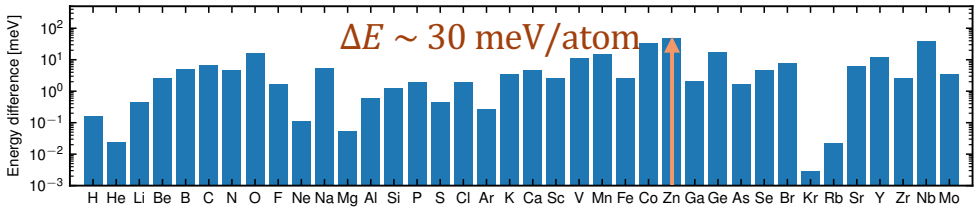


No molecular data



No low-dimensional structures

Inconsistent DFT settings

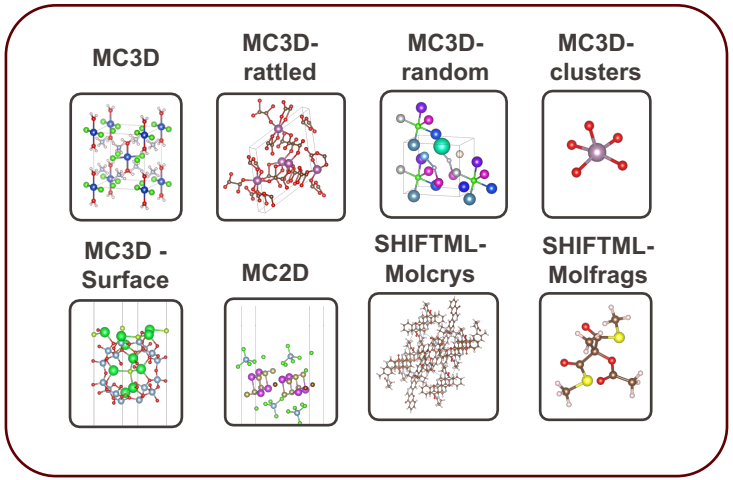


Energy difference between *consistent* and *default* DFT settings

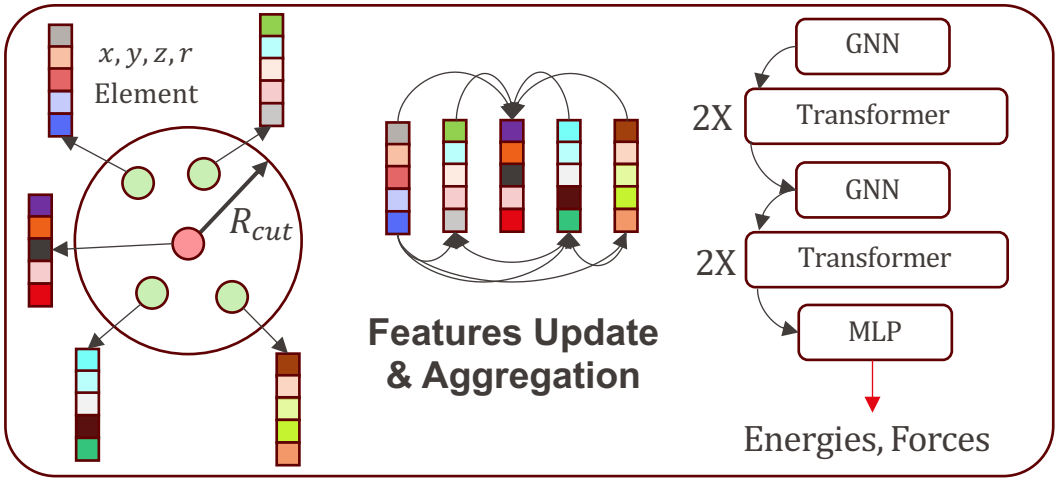
PET-MAD

A UNIVERSAL INTERATOMIC POTENTIAL

Massive Atomic Diversity (MAD) Dataset



Point-Edge Transformer (PET)



8 classes of systems with 85 elements, enhanced diversity and consistent DFT

Modern Transformer + GNN based architecture

Pozdnyakov, Sergey, and Michele Ceriotti. *Advances in Neural Information Processing Systems* 36 (2023): 79469-79501.

Mazitov, Arslan, et al. *arXiv preprint arXiv:2503.14118*(2025)

Features of PET-MAD

A UNIVERSAL INTERATOMIC POTENTIAL

- **Infinite** body-order
- **Unconstrained** architecture
- Learns **rotational equivariance** via Rotational Augmentations
- Has **non-conservative** (i.e. direct) **forces** and **stresses** prediction: 2x-3x speedup, yet a bit less accuracy
- Uncertainty **Quantification** and Uncertainty **Propagation** are available
- **ASE** and **LAMMPS** (with **KOKKOS** support) MD interfaces
- Fine-tuning capabilities

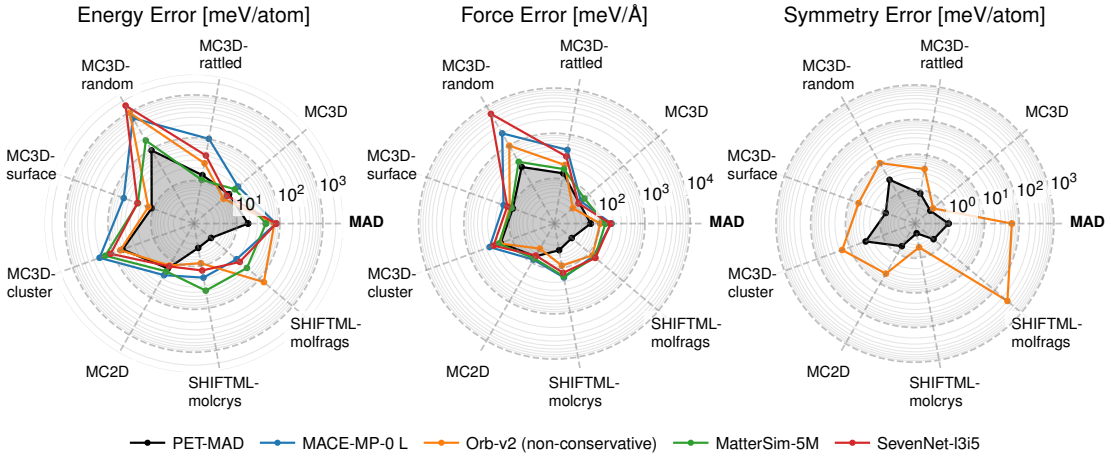
Mazitov, Arslan, et al. *arXiv preprint arXiv:2503.14118*(2025)

PET-MAD on benchmarks

Benchmarking on subsets of the MAD dataset against popular UMLIPS

WITH CONSISTENT DFT

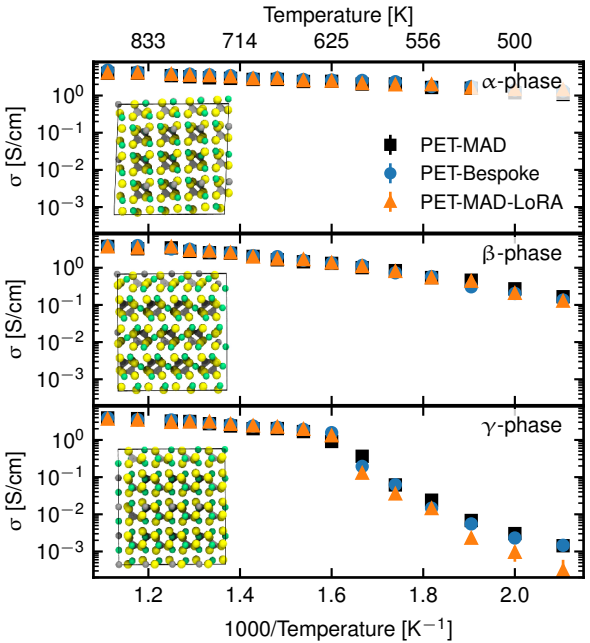
Benchmarking on various atomistic datasets



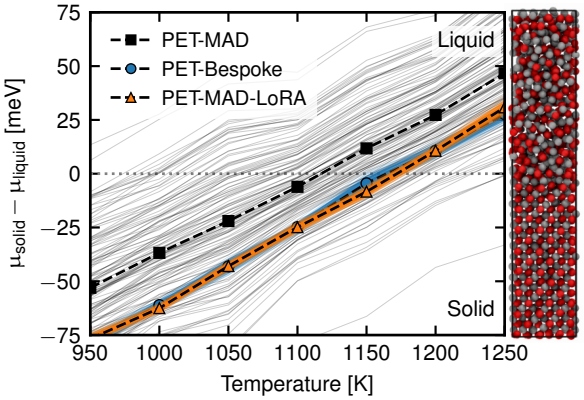
Dataset	PET-MAD	MACE-MP-0-L	MatterSim-5M	Orb-v2	SevenNet
MAD	**18.2**|**63.2**	81.6|181.5	47.3|133.7	74.4|103.0	82.1|173.5
MatBench	**45.8**|**66.6**	62.5|—	45.8|—	47.0|—	53.1|—
MPtrj	22.0|79.1	16.0|54.1	24.6|64.3	**5.5**|**25.1**	9.8|29.0
Alexandria	52.2|57.6	66.3|70.5	25.7|40.8	**11.6**|**9.1**	40.8|62.4
OC2020	**17.6**|114.6	82.8|167.8	30.9|118.1	20.6|**104.3**	46.4|157.5
SPICE	**3.7**|**59.8**	10.6|167.3	21.2|146.2	70.8|184.0	11.2|141.2
MD22	**1.9**|**65.8**	10.8|219.5	25.8|156.8	142.8|321.7	13.4|202.4

Mazitov, Arslan, et al. *arXiv preprint arXiv:2503.14118*(2025)

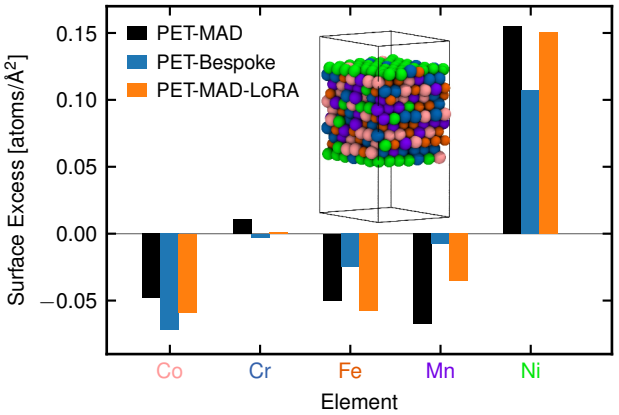
PET-MAD in applications



Ionic conductivity in LiPS4



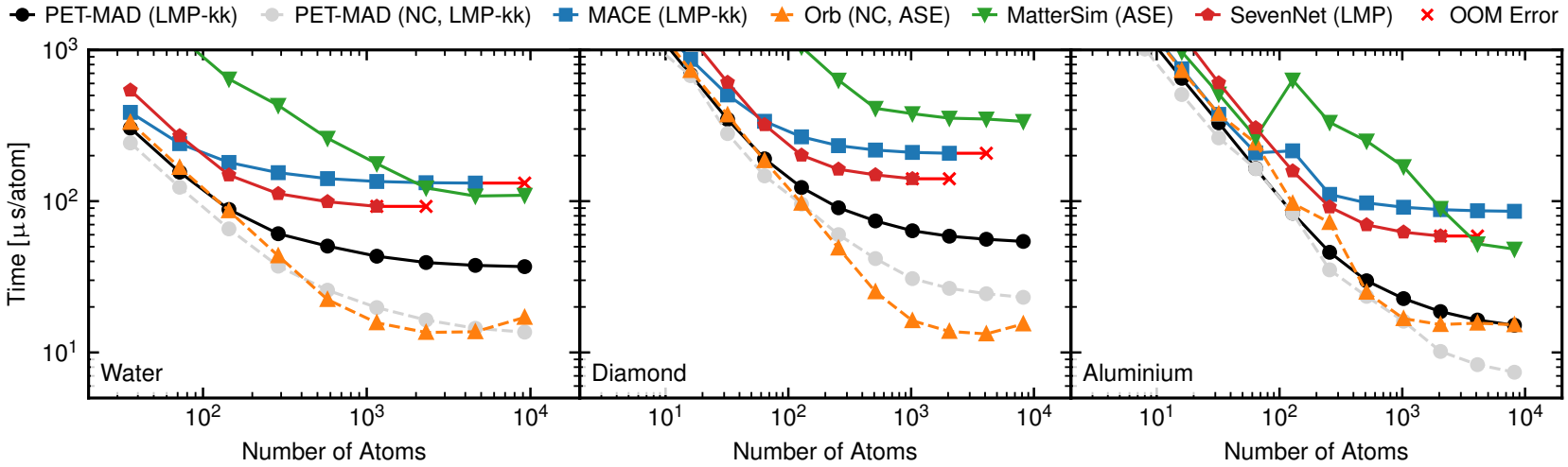
Melting point of GaAs



Surface segregation in CoCrFeMnNi HEA

Mazitov, Arslan, et al. *arXiv preprint arXiv:2503.14118*(2025)

PET-MAD inference speed



PET-MAD delivers SOTA speed in various systems!

Mazitov, Arslan, et al. *arXiv preprint arXiv:2503.14118*(2025)

PET-MAD in Metatensor ecosystem

PET-MAD is trained using Metatrain

etatrain

checks passing | codecov 91% | documentation latest

`metatrain` is a command line interface (cli) to `train` and `evaluate` atomistic models of various architectures. It features a common `yaml` option inputs to configure training and evaluation. Trained models are exported as standalone files that can be used directly in various molecular dynamics (MD) engines (e.g. `LAMMPS`, `i-PI`, `ASE` ...) using the metatensor atomistic interface.

The idea behind `metatrain` is to have a general hub that provide an homogeneous environment and user interface transforms every ML architecture in an end-to-end model that can be connected to an MD engine. Any custom architecture compatible with TorchScript can be integrated in `metatrain`, gaining automatic access to a training and evaluation interface, as well as compatibility with various MD engines.

metatensor/metatrain

Try PET-MAD!

PET-MAD

A UNIVERSAL INTERATOMIC POTENTIAL

PET-MAD: A Universal Interatomic Potential for Advanced Materials Modeling

This repository contains **PET-MAD** - a universal interatomic potential for advanced materials modeling across the periodic table. This model is based on the **Point Edge Transformer (PET)** model trained on the **Massive Atomic Diversity (MAD) Dataset** and is capable of predicting energies and forces in complex atomistic simulations.

lab-cosmo/pet-mad

Why Metatensor?

- Easy model training available via **Metatrain**
- Unified data interchange format via **Metatensor**
- Unified interface to atomistic simulation engines via **Metatomic**

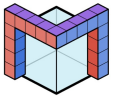
https://github.com/metatensor

etatrain

checks passing | codecov 91% | documentation latest

`metatrain` is a command line interface (cli) to `train` and `evaluate` atomistic models of various architectures. It features a common `yaml` option inputs to configure training and evaluation. Trained models are exported as standalone files that can be used directly in various molecular dynamics (MD) engines (e.g. `LAMMPS`, `i-PI`, `ASE` ...) using the metatensor atomistic interface.

The idea behind `metatrain` is to have a general hub that provide an homogeneous environment and user interface transforms every ML architecture in an end-to-end model that can be connected to an MD engine. Any custom architecture compatible with TorchScript can be integrated in `metatrain`, gaining automatic access to a training and evaluation interface, as well as compatibility with various MD engines.

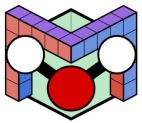


etatensor

checks passing | documentation latest | codecov 87%

Metatensor is a self-describing sparse tensor data format for atomistic machine learning and beyond; storing values and gradients of these values together. Think numpy `ndarray` or pytorch `Tensor` equipped with extra metadata for atomic systems and other point clouds data. The core of this library is written in Rust and we provide API for C, C++, and Python.

The main class of metatensor is the `TensorMap` data structure, defining a custom block-sparse data format. If you are using metatensor from Python, we additionally provide a collection of mathematical, logical and other utility operations to make working with TensorMaps more convenient.

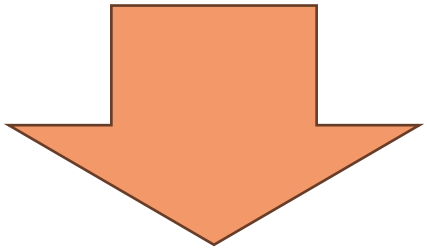


etatomic

checks passing | documentation latest | codecov 82%

`metatomic` is a library that defines a common interface between atomistic machine learning models, and atomistic simulation engines. Our main goal is to define and train models once, and then be able to re-use them across many different simulation engines (such as LAMMPS, GROMACS, *etc.*). We strive to achieve this goal without imposing any structure on the model itself, and to allow any model architecture to be used.

I tried PET-MAD, it didn't work on my data



Fine-tuning

Join the PET-MAD fine-tuning session today at 13:30 in MXF 312!

Big thanks to our team!

Filippo

Matthias

Paolo

Davide

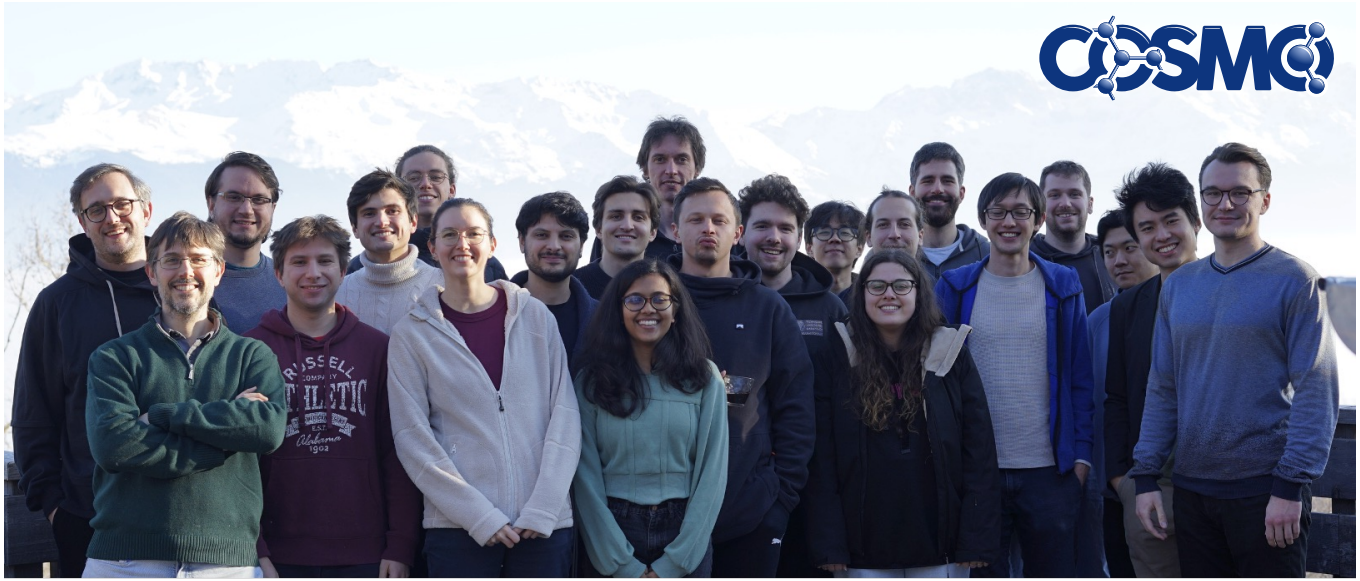
Guillaume

Sofia

Sergey

Michele

Big thanks to our group!



metatensor metatomic metatrain

- Software stack to meet the diverse needs of atomistic machine learning
- Developed with support from SNSF, ERC, and importantly: MARVEL

NATIONAL CENTRE OF COMPETENCE IN RESEARCH

Useful links

PET-MAD arXiv pre-print

https://arxiv.org/pdf/2503.14118

PET-MAD repo

lab-cosmo/pet-mad

Metatensor ecosystem

metatensor