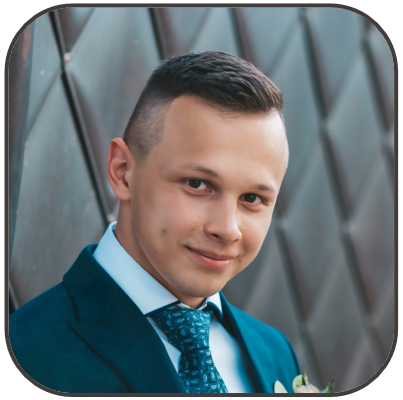
Tutorial Session:
Fine-tuning the PET-MAD universal potential for specific applications

Arslan Mazitov, Filippo Bigi, Hanna Türk
EPFL, Switzerland

Metatensor Workshop
June 13, 2025

Our Team



Arslan Mazitov
PhD student @
Lab COSMO

Main developer of PET-MAD

Filippo Bigi
PhD student @
Lab COSMO

Main developer of PET-MAD

Hanna Türk
Postdoc @
Lab COSMO

Plan

- **Introduction:** what is fine-tuning and how to use it?

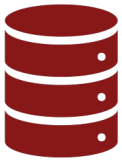
- **Tutorial session:**
  - Basic (full) fine-tuning
  - Heads fine-tuning
  - LoRA fine-tuning
  - Transfer learning

Fine-tuning in one minute

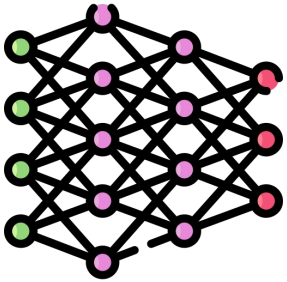
UMLIP

Works for arbitrary systems

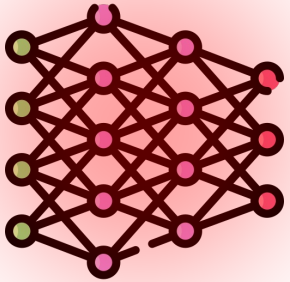
… maybe
… maybe not



New data



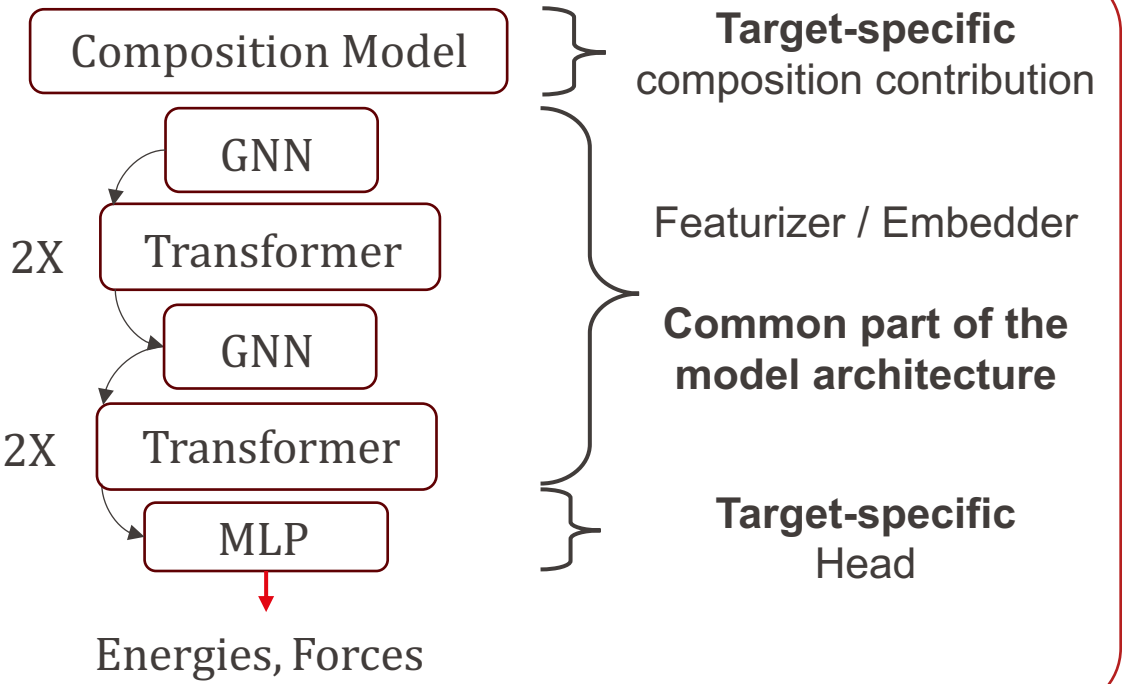
Pre-trained UMLIP



Fine-tuned UMLIP

Fine-tuning strategies

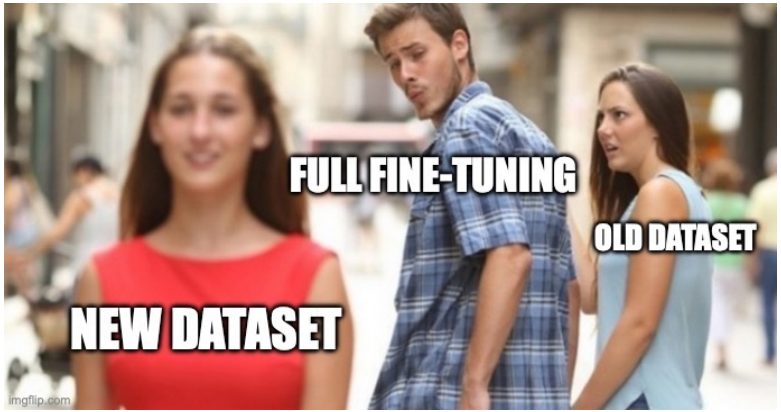
PET-MAD model architecture



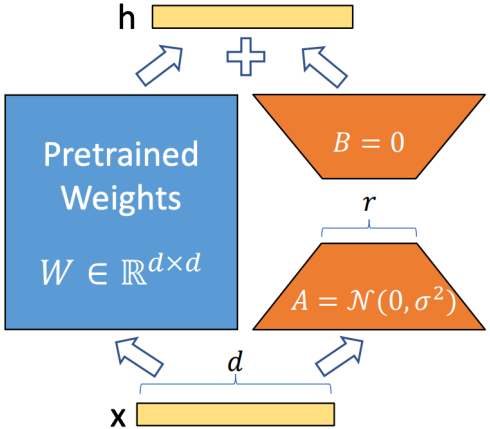
Fine-tuning strategies

1. Fine-tune **all the weights**
2. Fine-tune **selected parts of the model** (i.e. heads or composition model)
3. Add **new weights** and fine-tune them only (LoRA)
4. Add new targets (i.e. new target-specific weights)

Low-Rank Adaptation (LoRA)



Fully fine-tuned model can "forget" the original dataset



$$W = W_0 + \frac{\alpha}{r} AB$$

LoRA fine-tuning allows to balance between two

In theory. Does it really work like this? We will see.

Getting started


```
git clone
https://github.com/metatensor/
Workshop-spring-2025.git

cd finetuning/

python3 -m venv virtualenv
source ./virtualenv/bin/activate
pip install -U pip

pip install -r requirements.txt
```

Quick overview of the examples and scripts

Structure of the folders

```
00.initial-evaluation
01.full-finetuning
02.heads-finetuning
03.lora-finetuning
04.transfer-learning

shared/
     datasets
     models
scripts
```

Runs the finetuning

finetuning.sh

```
mtt train options.yaml
    -o model.pt
```

Evaluates the fine-tuned model

eval.sh

```
mtt eval model.pt
eval-options.yaml -o
predictions.xyz
```

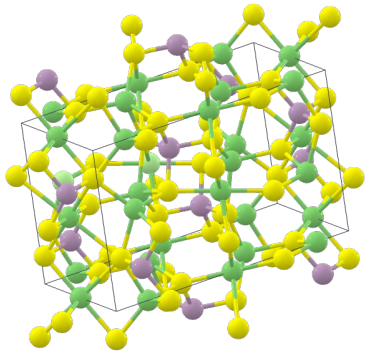
Model and dataset

Model: PET-MAD v1.0.1

PET-MAD

A UNIVERSAL INTERATOMIC POTENTIAL

Dataset: Ionic conductivity in Li3PS4



`shared/datasets/Li3PS4-sample`

How to get a model?

Hugging Face

https://huggingface.co/lab-cosmo/pet-mad

Quick Start (for applications)

```
mtt export https://huggingface.co/lab-cosmo/pet-mad/resolve/v1.0.1/models/pet-mad-latest.ckpt
```

Fetches the checkpoint and converts to TorchScript format for applications

Get a checkpoint for fine-tuning

```
wget https://huggingface.co/lab-cosmo/pet-mad/resolve/v1.0.1/models/pet-mad-latest.ckpt
```

Downloads the raw checkpoint

Export .ckpt to TorchScript

```
mtt export pet-mad-latest.ckpt
```

Export a local .ckpt to TorchScript

Fine-tuning steps

1. Run `finetuning.sh` to **fine-tune** the model

2. Run the `eval.sh` to **evaluate** the **performance** of the fine-tuned model on Li3PS4 and MAD datasets

3. **Examine** the performance using the Jupyter Notebook `inspect_errors.ipynb`

Structure of the options.yaml


```yaml
base_precision: 32 # float 32
seed: 0 # fix random seed
architecture:
  name: pet
  model:
    d_pet: 256 # Latent space dim

training:
  batch_size: 8
  num_epochs: 20
  num_epochs_warmup: 0
  checkpoint_interval: 5
  learning_rate: 1e-5
  finetune:
    method: "full" # heads / lora
    read_from: "../shared/models/
pet-mad-v1.0.1.ckpt"
```

```yaml
training_set:
  systems:
    read_from: "../shared/datasets/
Li3PS4-sample/train.xyz"
    length_unit: angstrom
  targets:
    energy:
      unit: eV
      forces: false
      stress: false

validation_set:
...

test_set:
...
```

How to set fine-tuning in options.yaml

Full

```yaml
training:
  finetune:
    method: "full"
    read_from: "../shared/models/pet-mad-v1.0.1.ckpt"
```

Heads

```yaml
training:
  finetune:
    method: "heads"
    read_from: "../shared/models/pet-mad-v1.0.1.ckpt"
    config:
      head_modules: ['node_heads', 'edge_heads']
      last_layer_modules: ['node_last_layers', 'edge_last_layers']
```

LoRA

```yaml
training:
  finetune:
    method: "lora"
    read_from: "../shared/models/pet-mad-v1.0.1.ckpt"
    config:
      alpha: 0.1
      rank: 4
```

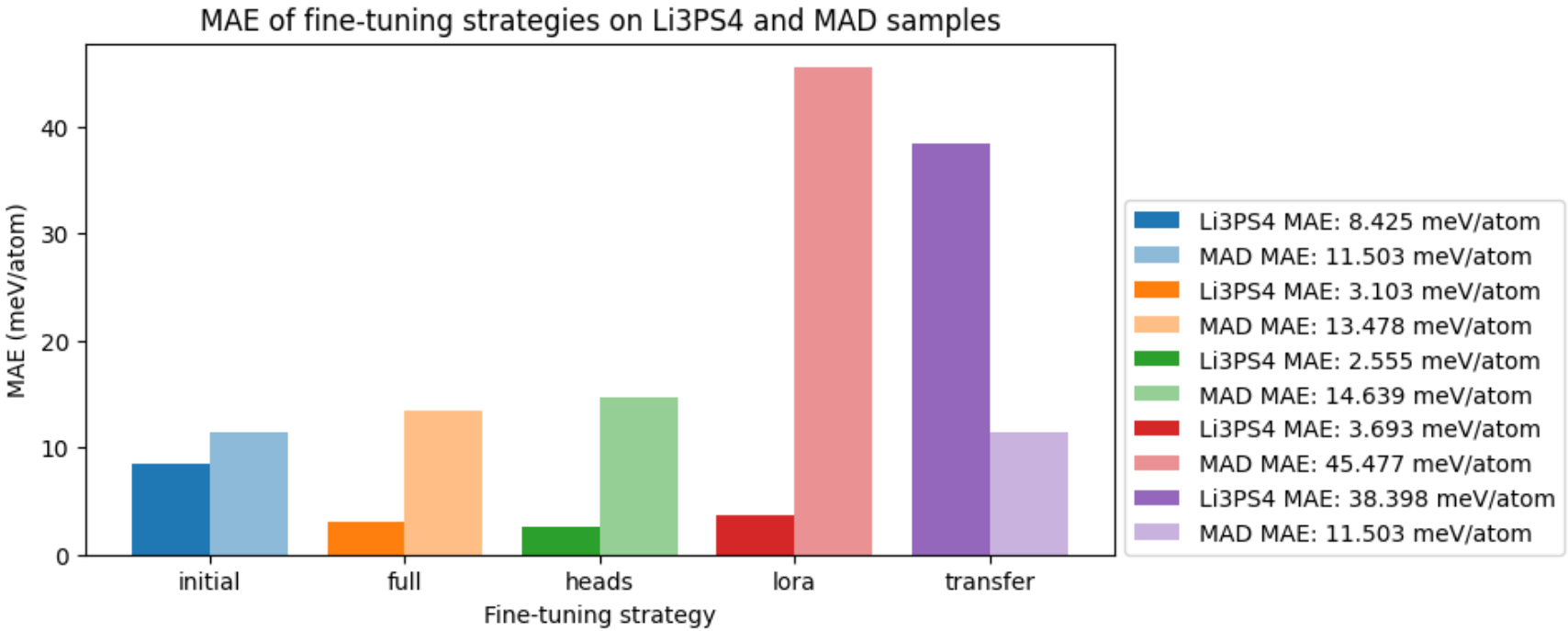
How to set transfer learning in options.yaml

```yaml
training:
  finetune:
    method: ”heads”
    read_from: "../shared/models/pet-mad-v1.0.1.ckpt”
    config:
      head_modules: ['node_heads', 'edge_heads’]
      last_layer_modules: ['node_last_layers', 'edge_last_layers’]

training_set:
  systems:
    read_from: "../shared/datasets/Li3PS4-sample/train.xyz"
    length_unit: angstrom
  targets:
    mtt::r2scan_energy:
      unit: eV
      forces: false
      stress: false
```

We are combining new heads fine-tuning with the new r2scan energy target!

Fine-tuning strategies summary



Useful links

PET-MAD arXiv pre-print

https://arxiv.org/pdf/2503.14118

PET-MAD repo

lab-cosmo/pet-mad

Metatensor ecosystem

metatensor