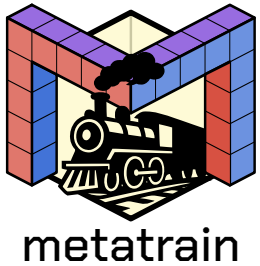


Metatrain

a tool to train your own models – without writing code

Filipo Bigi & Philip Loche — Metatensor workshop June 2025

What is Metatrain

Machine learning for atoms – made easy.

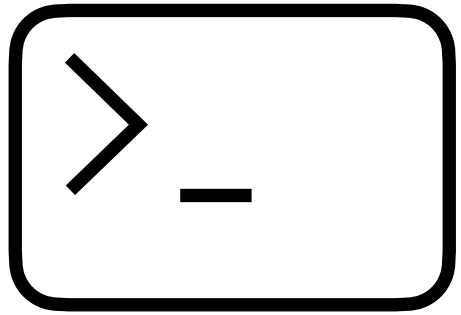
✔ Train models

✔ Evaluate models

✔ Export models for simulations etc.

Turn complex ML into simple commands.

No deep ML knowledge required –
configure with YAML, run from the terminal.



```
architecture:
  name: soap_bpnn

tra
val
tes
```

```
architecture:
  name: gap

trair
valid
test_
```

```
architecture:
  name: pet

training_set: water.xyz
validation_set: 0.1
test_set: 0.1
```

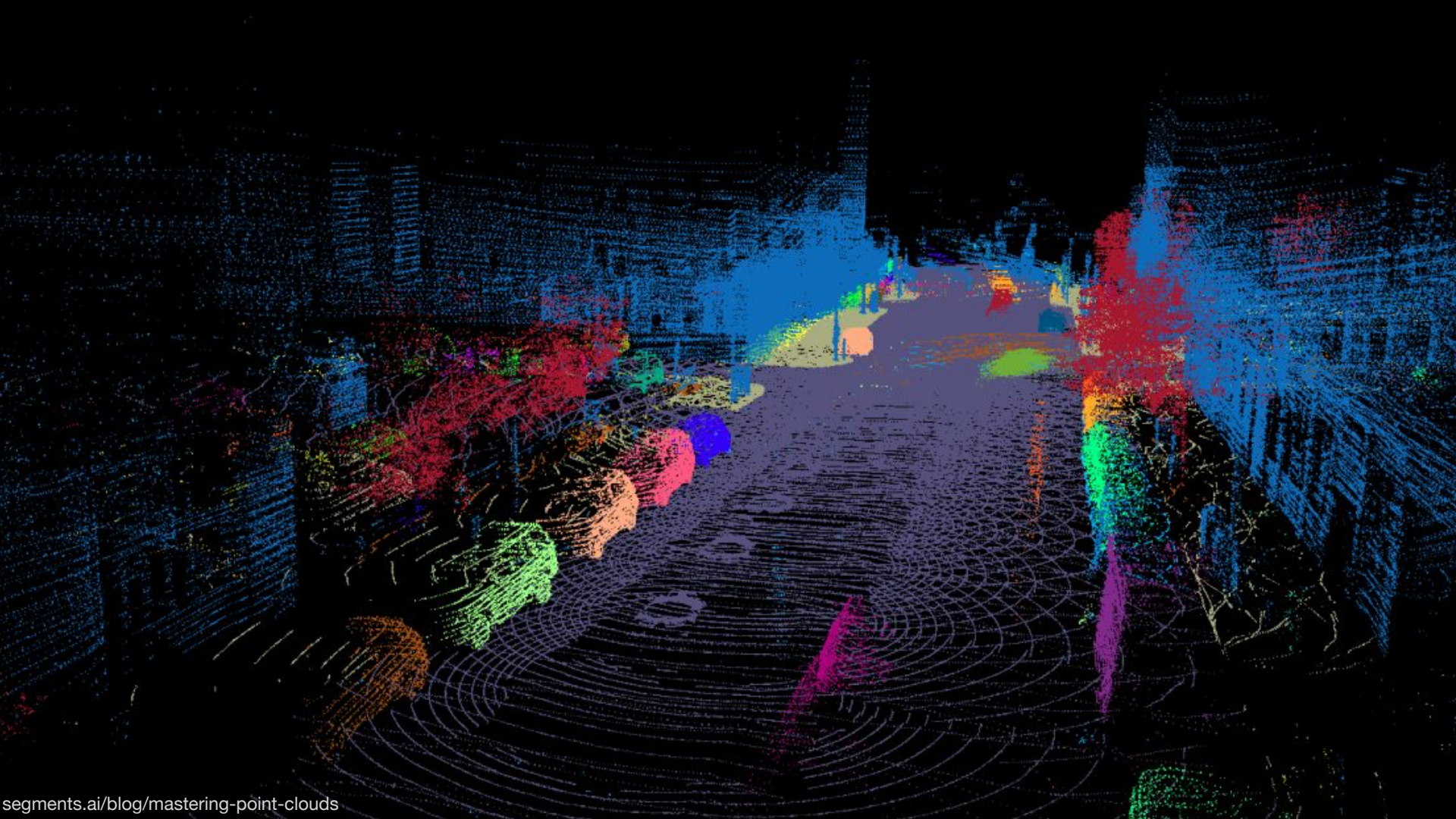
Key Features

📝 Simple config via YAML

🖥️ fully scriptable and CLI-based

🏗️ Architecture-agnostic

⚙️ Single interface for training & inference



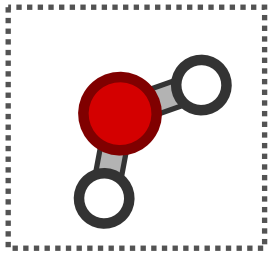
segments.ai/blog/mastering-point-clouds

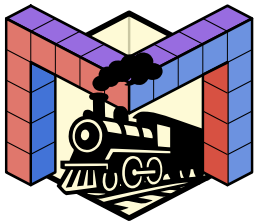
Our flavor of data driven modeling

Supervised learning for **regression** and (classification) based on **colored point clouds**

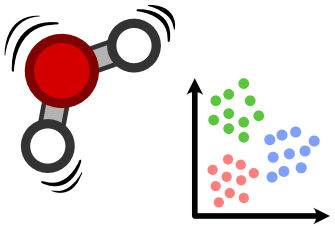
Features:
Cell, elements, positions

Targets:
Energies, forces, dipole moments, …





metatrain



```
cell: 1.0 1.0 1.0

H 0.00 0.11 0.32
H 0.24 0.11 0.21
O 0.13 0.21 0.15
...
```

```
energy: 1.023

H -0.12  0.24 -0.11
H -0.42 -0.56 -0.16
O -0.54  0.13  0.86
```

Supported Architectures

🏗️ **Architecture**: The **blueprint**
→ how model is built/trained (features, network, losses, optimizer)

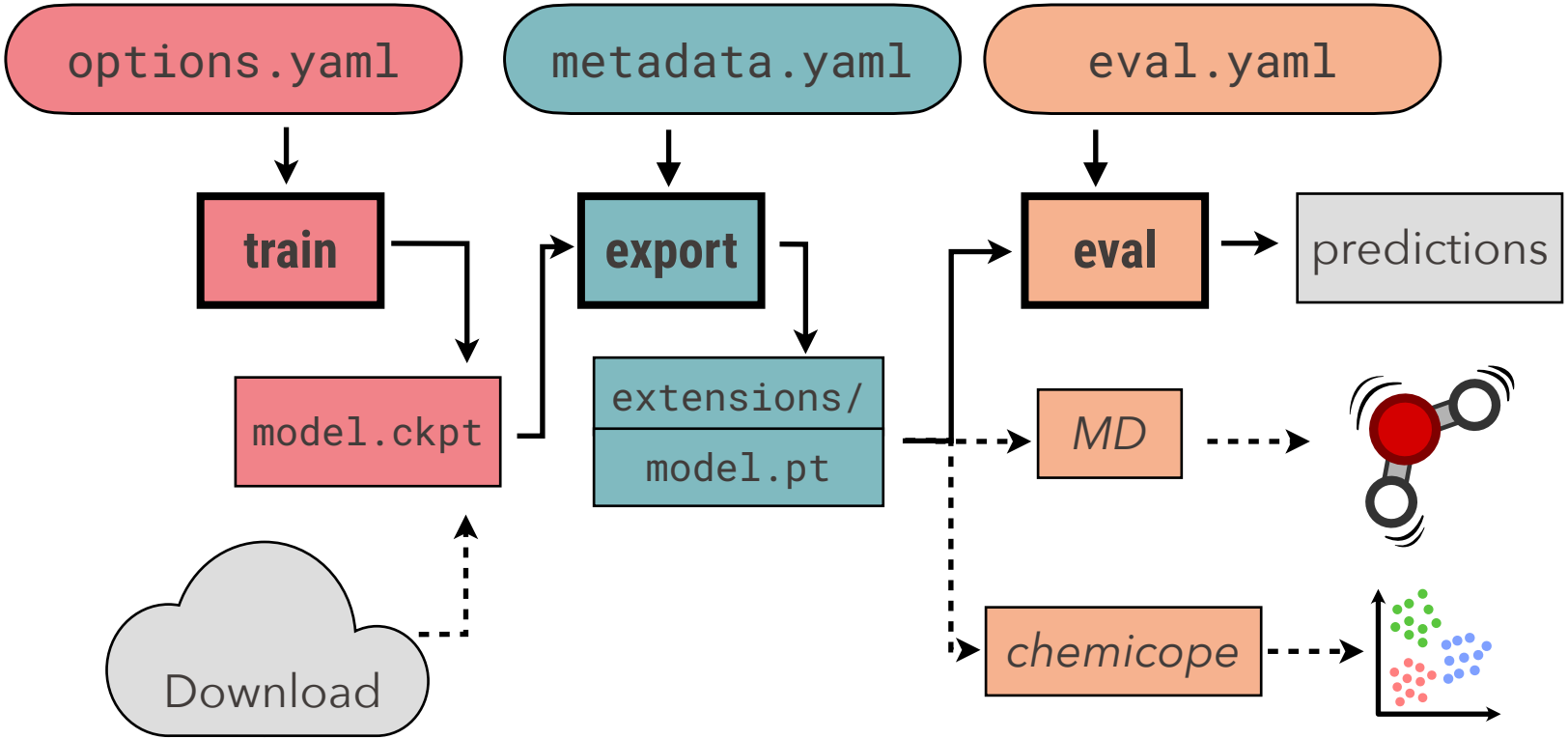
📦 **Model**: The **finished product**
→ a trained, ready-to-use *file* for predictions or simulations

📊 What can the models predict?

Model	Energy & its gradients	Scalars	Spherical tensors	Cartesian tensors
GAP	✔ Energy, Forces	✔	❌	❌
SOAP-BPNN	✔ Energy, Forces, Stress	✔	✔	❌
PET	✔ Energy, Forces, Stress	✔	✔	✔ Vectors & 2D Tensors

🚀 *Metatrain Workflow*

From configuration to simulation in one seamless flow.



🏋️ *Training Your Model*

Metatrain needs:

📂 Dataset → your atomic structures

 Options file → options.yaml with your training setup

You get:

📦 model.pt → final trained model

💾 Checkpoints along the way (model.ckpt)

📊 Training logs & stats

🔀 Dataset splits (train/validation/test)

```
mtt train options.yaml
```

DEMO

✅ *Why train with metatrain?*

📝 Full YAML input – easy to write, easy to version, industry standard

🔀 Automatic dataset splitting – reproducible with fixed seed

⚡ Shell completion – speed up your workflow with tab-completion

🎯 Multi-target training – predict multiple properties at once

⚙️ Good defaults, fully customizable – no black box

🛠️ Helpful error messages – we try to be nice

📡 W&B integration – track experiments like a pro

🚀 Distributed training – ready for big compute

📖 Readable logs – for humans and machines

🧐 *Evaluating Your Model*

Metatrain needs:

📦 Model → your trained model (model.pt)

📂 Dataset → systems to evaluate on

 Options file → eval.yaml with evaluation setup

You get:

📈 Predictions – energies, forces, tensors, …

📉 Errors – detailed metrics for each target

```
mtt eval model.pt eval.yaml
```

DEMO

📦 *Exporting Your Model*

Metatrain needs:

💾 Checkpoint → local file or remote location

📝 Metadata file → metadata.yaml with your references

HUGGING FACE

You get:

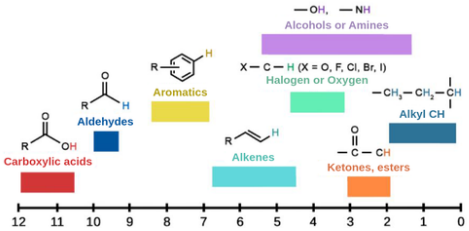
🚀 Exported model → ready for production in MD engines (LAMMPS, i-PI, ASE…)

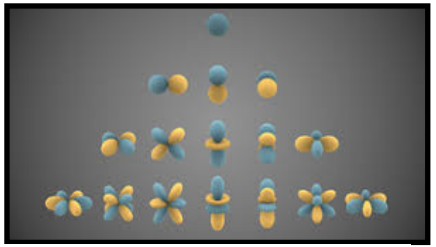

```
mtt export model.ckpt
```

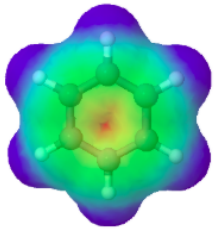
DEMO

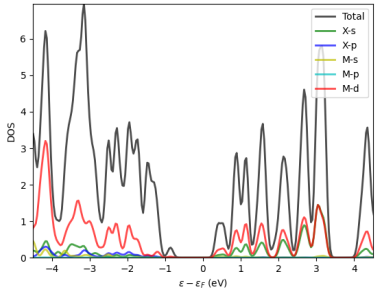
What Can You Train with Metatrain?

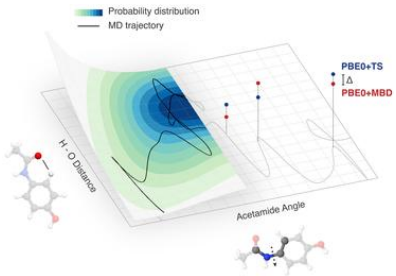
- 🧪 Chemical shifts
- 🔄 Equivariant properties (vectors, tensors, …)
- 🗺️ Electron densities
- 🧮 Hamiltonian matrices
- 📊 Density of states
- 🎥 Molecular dynamics trajectories
- ⚛️ Universal interatomic potentials
- ❓ Uncertainty-aware models

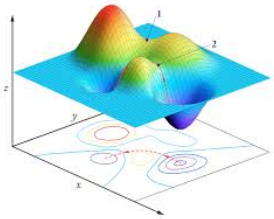












Try it, break it

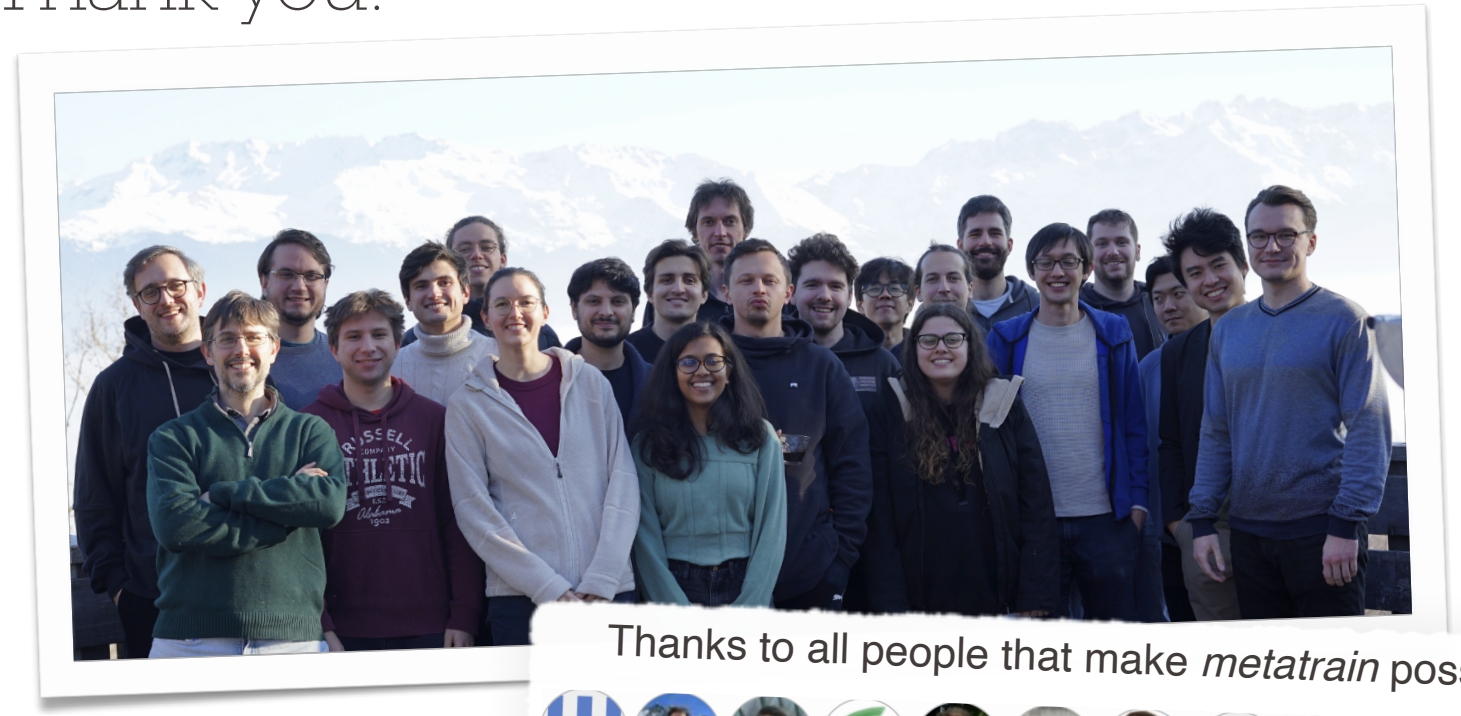
One command. Ready to train.

```
pip install metatrain
```

```
conda install \
  -c metatensor -c conda-forge \
  metatrain
```

metatensor/metatrain

Thank you!



Thanks to all people that make *metatrain* possible