

From Neurons to Atoms: How PyTorch Powers Open Science at Scale

EMPOWERING POTENTIAL

Elisa Cascardi
PgM, FAIR

Ray Gao,
Research Engineer, FAIR

Hubert Banville
Research Scientist, FAIR

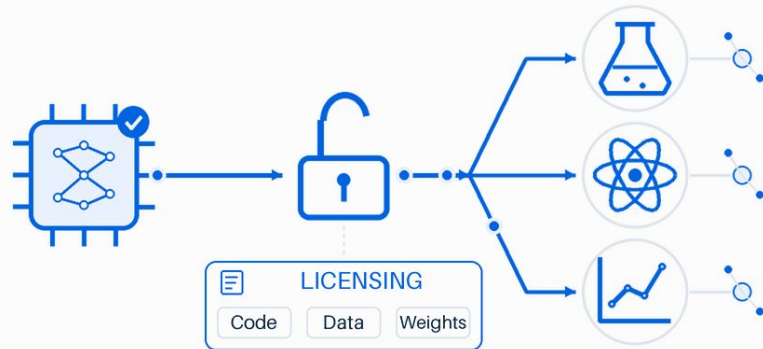


Open source is our design decision from day 1

Model success, trapped by infra



Open infra enables continued discovery



Open science is enabling discoveries at scale

UNIVERSAL MODEL FOR ATOMS (UMA) / CHEMISTRY

7,600+

unique accepted researchers

2,296

FAIRChem github stars

NEURALSET / NEUROSCIENCE

120,170

Total pip installs

328

NeuroAI github stars

Each release becomes a starting point for the next breakthrough

How can we enable community breakthroughs?

01 BUILD WITH OPEN ARTIFACTS

UMA / FAIRChem

PyTorch · BSD-3-Clause
Datasets · Software · Code

NeuralSet / Brain and AI

pandas · BSD-3-Clause
Pydantic / Exca · MIT
Datasets · Software · Code

02 RESEARCH & RISK MITIGATIONS

Model Development
and Testing + CBRNE
/ Dual Use Mitigations
+ Safety gating

Pipeline Development
and Testing + Risk
Mitigations

03 DEFINE TERMS & RELEASE

FAIRChem code | UMA Weights | OMol Data
Commercially permissive licensing

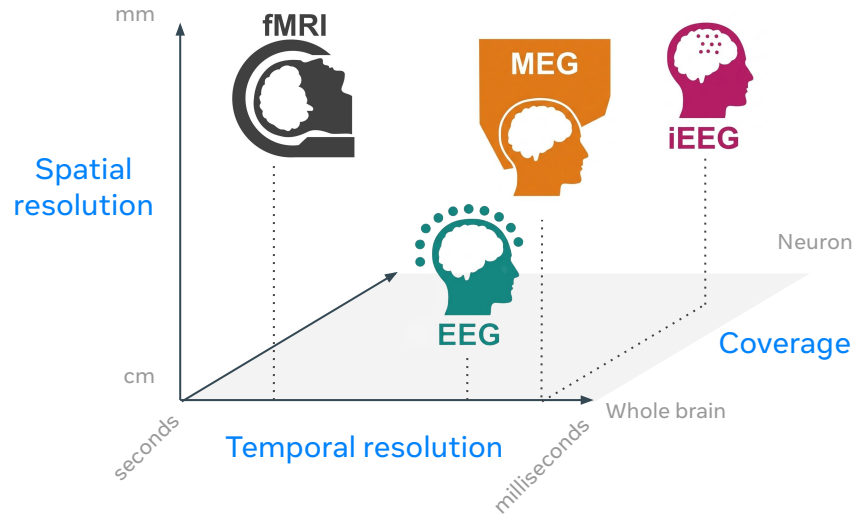
NeuralSet
Commercially permissive license

NEUROSCIENCE

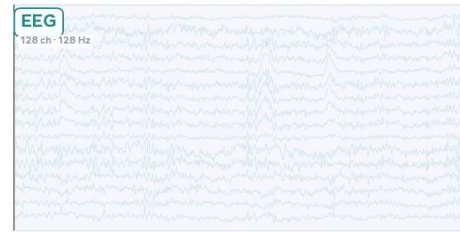
Modeling the brain with Pytorch

The neuroai stack

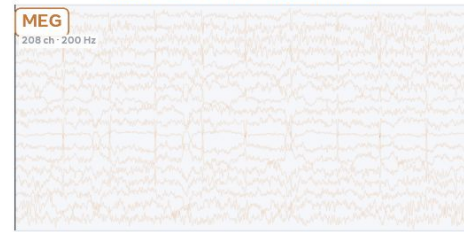
Brain data is messy – many devices, resolutions and formats



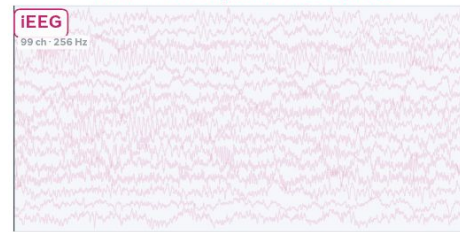
EEG: cheap & portable, great timing — but low SNR



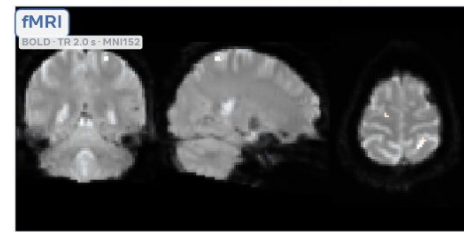
MEG: great SNR & timing — but costly and fixed



iEEG: excellent SNR & spatial resolution — but invasive



fMRI: whole-brain mm detail — but slow dynamics



...and it must be aligned with the world – what people see and hear

MEG

1,000 samples / s

fMRI

1 volume / 1.5 s

Images

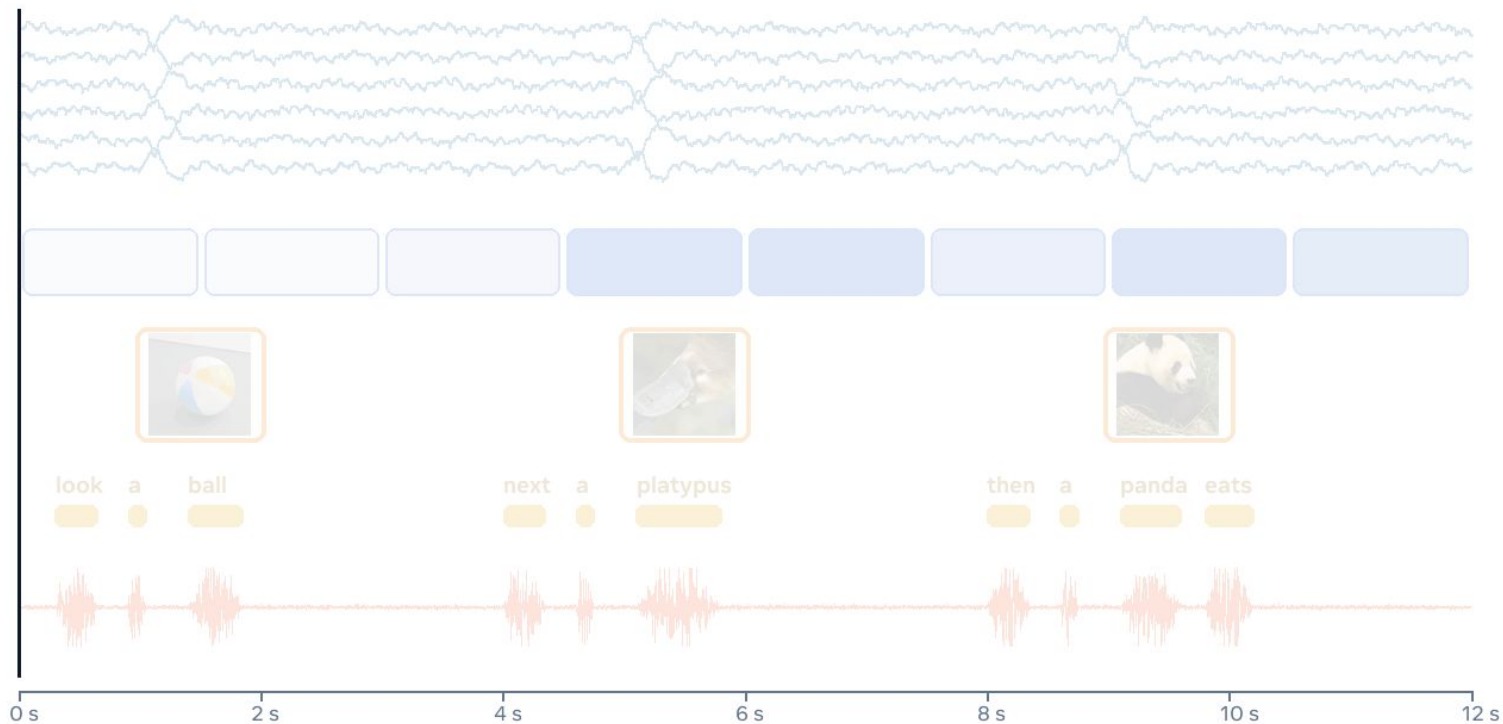
shown on screen

Words

heard

Audio

16,000 samples / s



Two modeling directions – Decoding and encoding



Challenges

- Support **multiple brain imaging devices**
- Support **multiple stimulus modalities** and event types
- Handle CPU-heavy preprocessing + support flexible (re)configuration
- **Work at scale:** from tens of subjects → 10k+ subjects, 100k+ hours

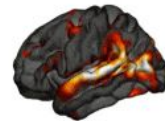
Seen image



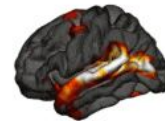
Decoded image



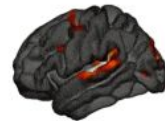
Predicted
put them in your purse



True
put them in your purse



Predicted



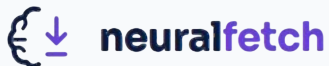
True



Decoding: images reconstructed from brain activity

Encoding: brain responses predicted from a movie

neuroai – an open-source PyTorch stack for brain modeling



100+ curated datasets



events → torch
DataLoader



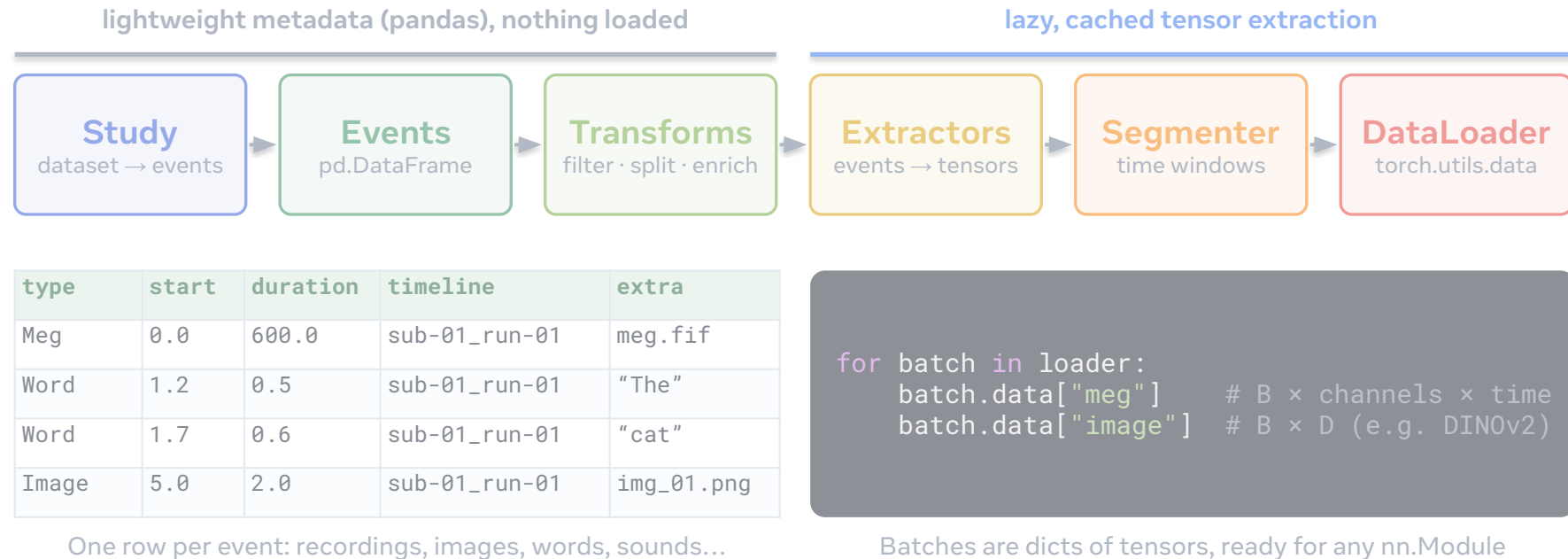
Deep learning building
blocks



Open benchmarking of
brain models

```
pip install neuralset
```

Lazy by design: metadata until the DataLoader asks for tensors



Each step is a pydantic config, cached by **exca** → from laptop to SLURM with one flag

Swap the device or the stim model by changing one line

```
import neuralset as ns
from torch.utils.data import DataLoader

events = ns.Study(name="MyStudy2026", path=DATA).run() # pandas

brain = ns.extractors.MegExtractor(frequency=120)
#      EegExtractor · FmriExtractor · IeegExtractor · FnirsExtractor · EmgExtractor ...
stim = ns.extractors.HuggingFaceImage(model_name="facebook/dinov2-small")
#      HuggingFaceText · HuggingFaceAudio · HuggingFaceVideo · Whisper ...

segmenter = ns.Segmenter(start=-0.1, duration=0.5, trigger_query='type=="Image"',
                          extractors=dict(brain=brain, stim=stim))
dataset = segmenter.apply(events) # torch.utils.data.Dataset
loader = DataLoader(dataset, batch_size=32, collate_fn=dataset.collate_fn)
```

Same code, any of 8 brain devices × any HuggingFace model. Interactive builder: facebookresearch.github.io/neuroai

Built for scale – neuroai in numbers

DATA

100+

curated datasets, one-line
download

24k

subjects

45k+

hours of recordings

8

brain devices supported:
EEG, MEG, fMRI, iEEG,
fNIRS, EMG, ECG, spikes

ENGINEERING

43

extractors, incl. any
HuggingFace image, text,
audio & video model

~40 μ s

framework overhead per
sample (`__getitem__`)

~2 kB

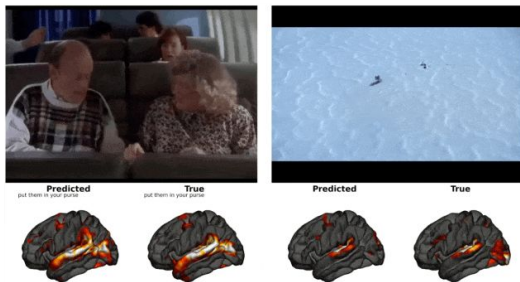
per batch of 32 sent to
workers: metadata, not
arrays

3.3 TB

of raw EEG through the
full benchmark

Powering our open source research at FAIR

Live demo at the booth



TRIBE v2

Predicts fMRI brain responses to movies, sound and language



Brain2Qwerty v2

Decodes typed sentences from non-invasive MEG

36

EEG tasks

94

EEG datasets

9.5k

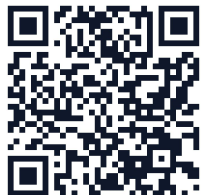
subjects

14

models



Open benchmark for brain models on EEG, MEG & fMRI



github.com/facebookresearch/neuroai

`pip install neuralset neuralfetch neuraltrain neuralbench`
MIT licensed; built on PyTorch, exca, MNE, nilearn & HuggingFace

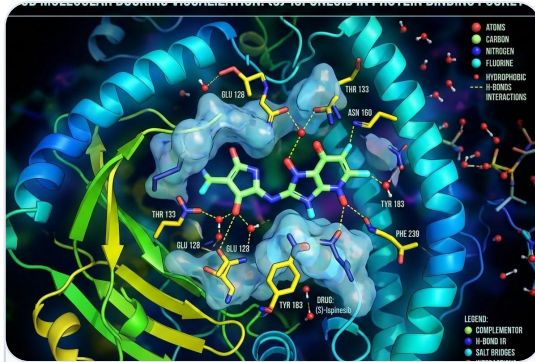


CHEMISTRY

From neurons to atoms

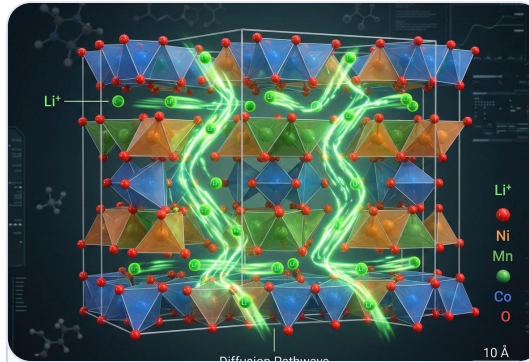
Universal models for open science

Density functional theory – one framework, many domains



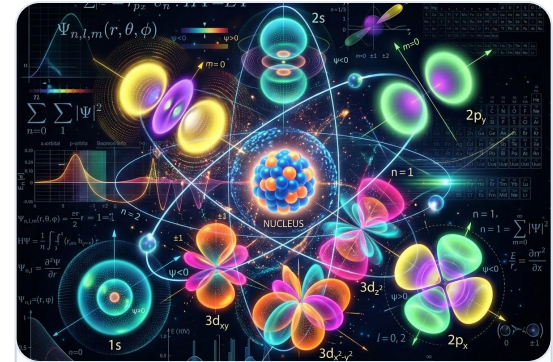
Biochemistry

- Drug discovery
- Enzyme reaction pathways
- Molecular geometry



Materials Science

- Battery ion transport
- 2D materials & graphene
- Polymer dynamics



Condensed Matter Physics

- Electronic band structures
- Superconductivity mechanisms
- Magnetism & spin state analysis

Quantum chemistry – high accuracy, high computational cost

First-principles electronic structure calculations

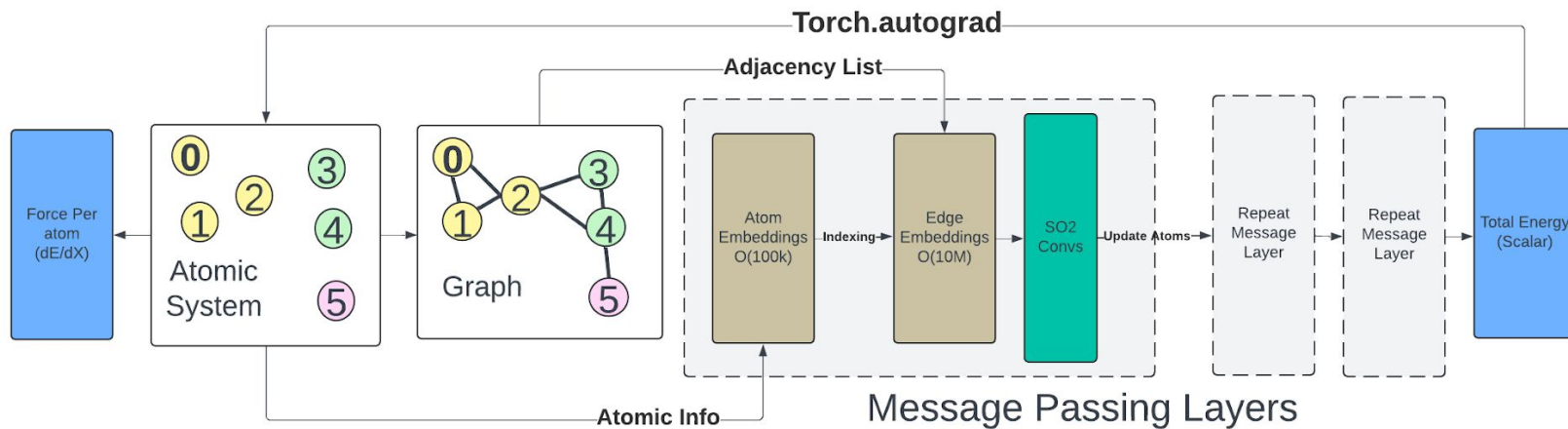


Learned interatomic potentials – fast neural-network surrogates

Predict energies and per-atom forces from atomic structures



Training UMA – graphs, forces and double backprop



Double backprop

Slow and memory intensive

Large graphs

Typically ~100 edges per node

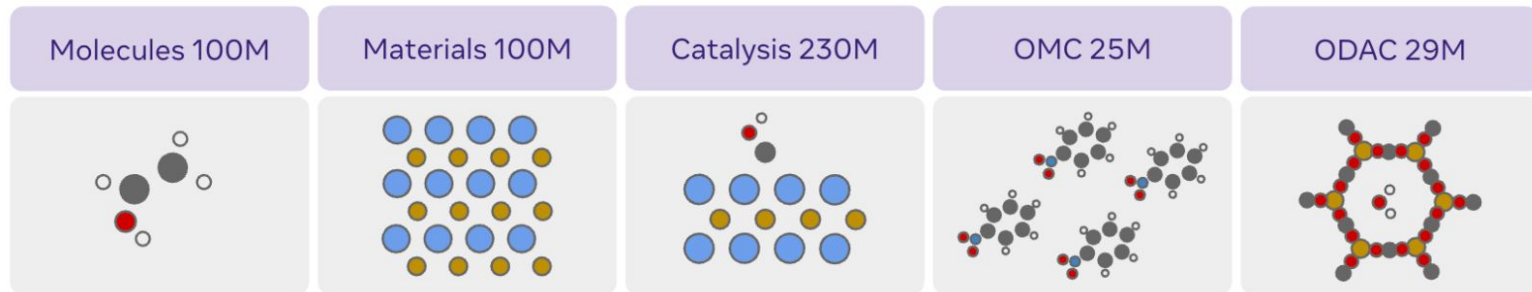
Dynamic inputs

Changing graphs and wide ranges

MoLE memory

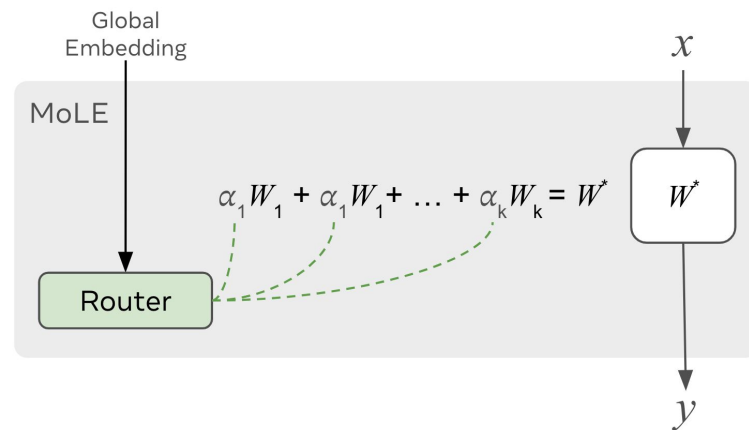
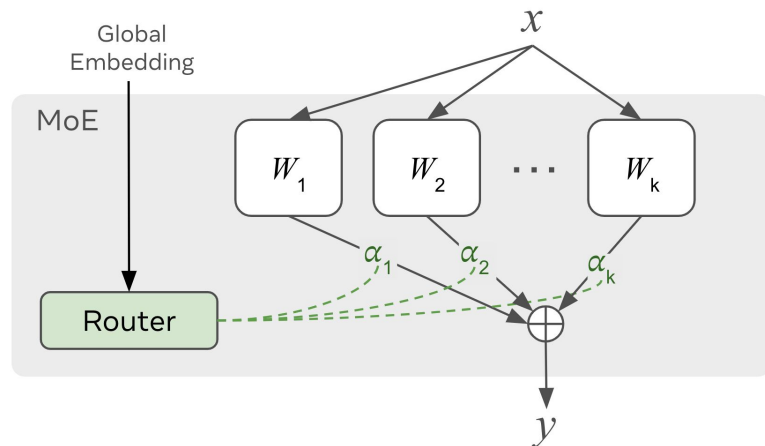
Large expert parameter storage

MoLE – increase capacity without sacrificing speed



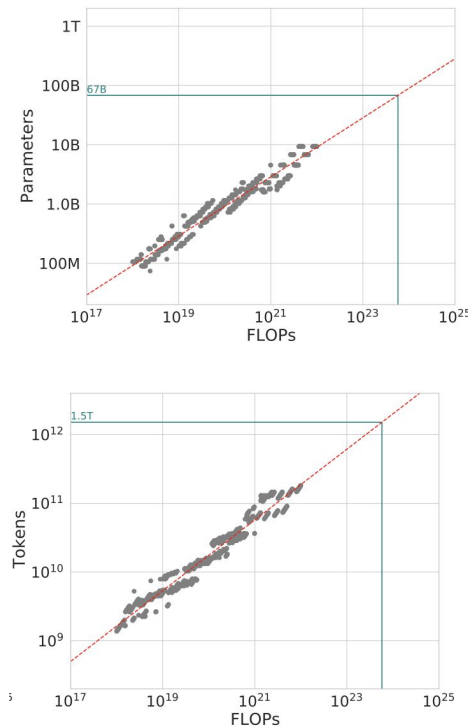
MIXTURE OF EXPERTS

MIXTURE OF LINEAR EXPERTS



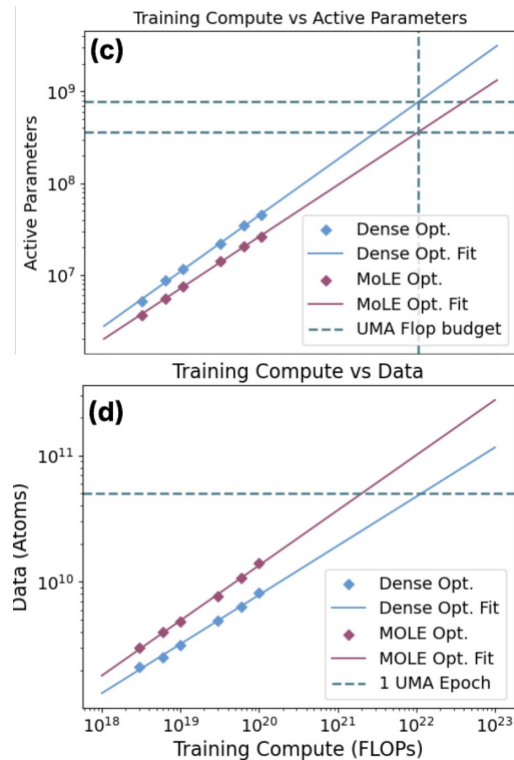
Scaling laws – larger datasets need larger models

LANGUAGE MODELS



Hoffmann et al. (2022), Training compute-optimal large language models.
arXiv:2203.15556

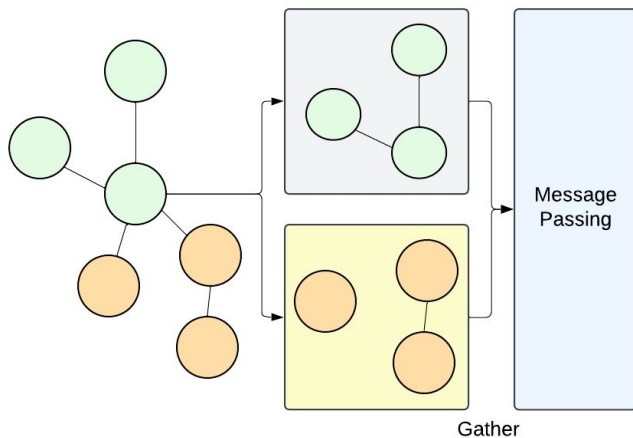
UNIVERSAL MODELS FOR ATOMS



Wood et al. (2025), UMA: A Family of Universal Models for Atoms.
arXiv:2506.23971

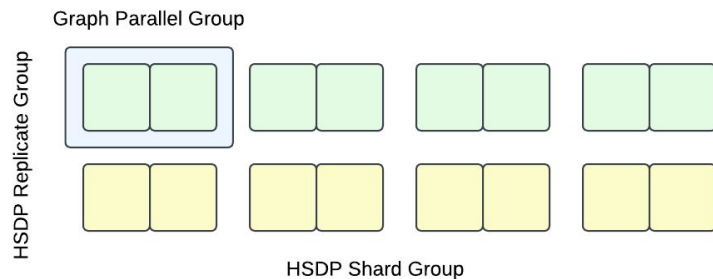
Distributed training – combine three parallelisms

GRAPH PARALLEL



Split graph computation across GPUs

DEVICE MESHES + FSDP2

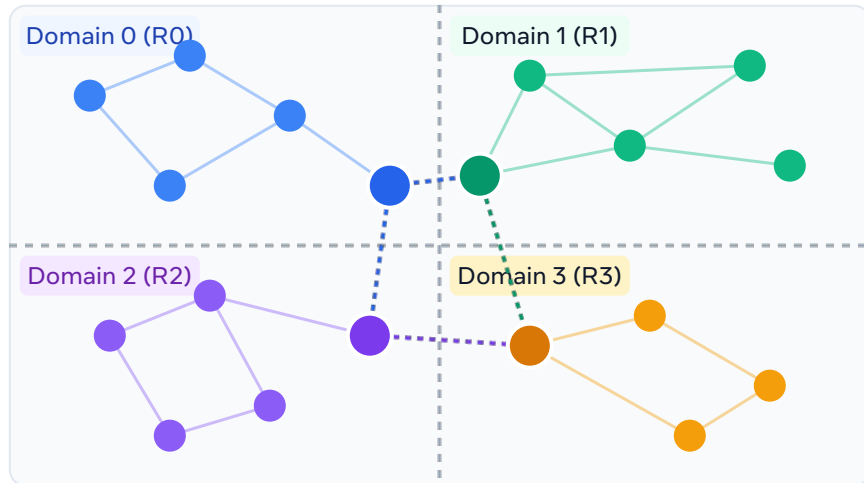


FSDP + graph parallel + DDP

Double backprop across multiple parallelisms makes training a challenging systems problem.

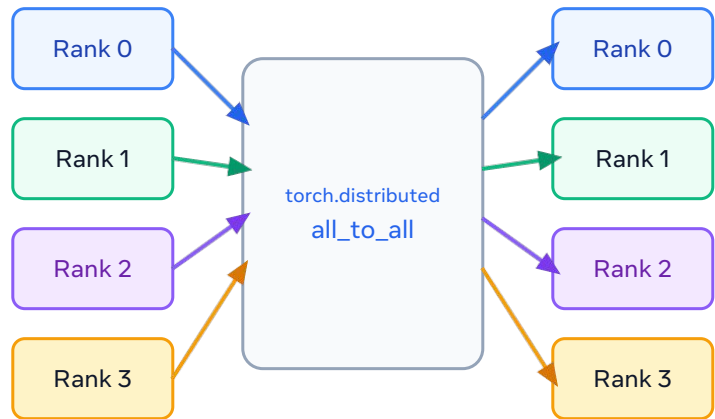
Graph parallel inference – split space across GPUs

1. Spatial domains and message passing



Assign spatial domains to GPU ranks. Exchange boundary embeddings during forward and backward passes.

2. Dynamic all-to-all exchange



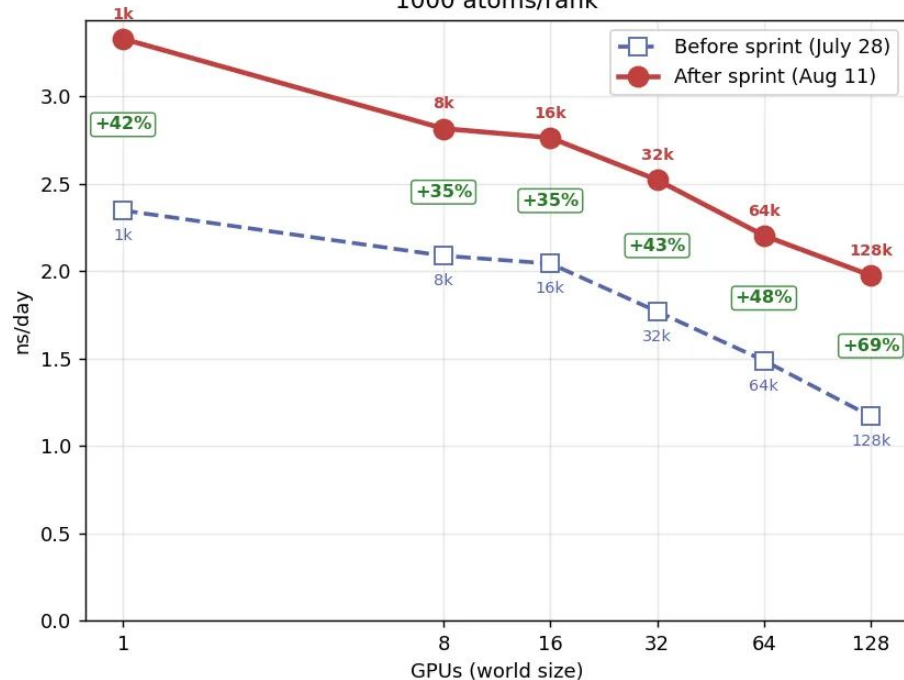
Exchange uneven tensor shapes directly across ranks, without fixed-buffer padding.

Local graph computation + dynamic `torch.distributed.all_to_all` = scalable inference.

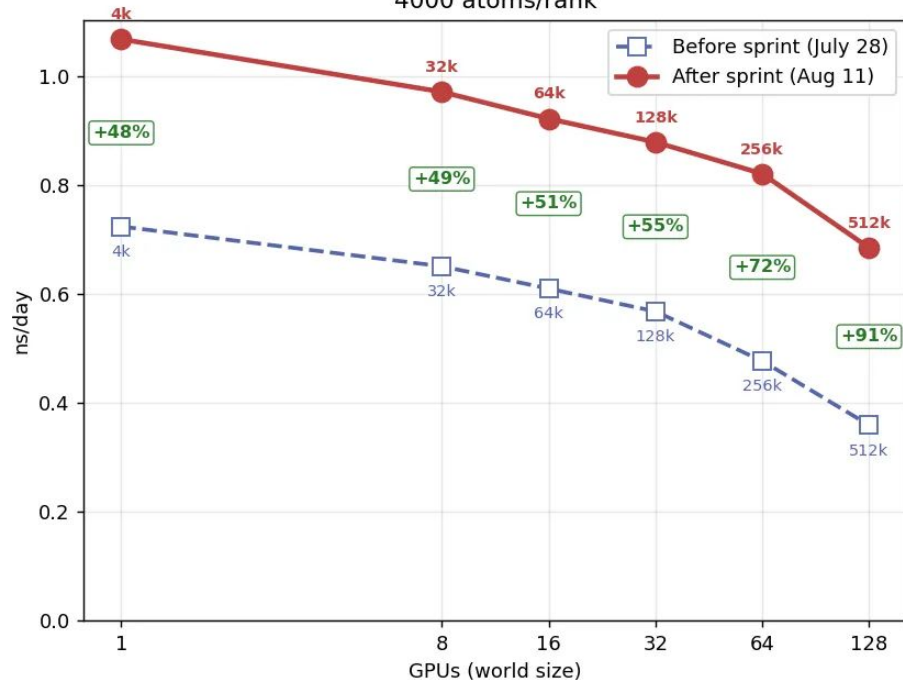
Collaboration with Pytorch team

Inference speed on H200 Turbo

1000 atoms/rank

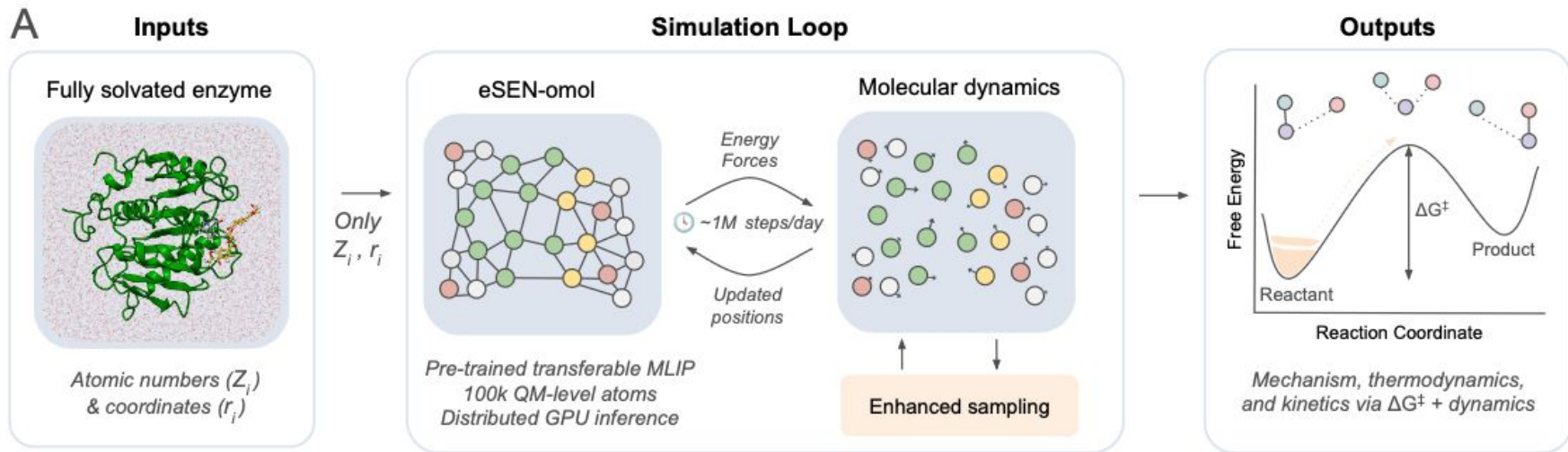


4000 atoms/rank



Quantum enzyme simulations – complete systems in solvent

eSEN-omol enables quantum-accurate, all-atom simulations in explicit solvent



54k

atoms in complete enzyme systems

1 μ s

total simulation time

Quantum-accurate atomistic modeling of enzyme catalysis using a machine learned potential

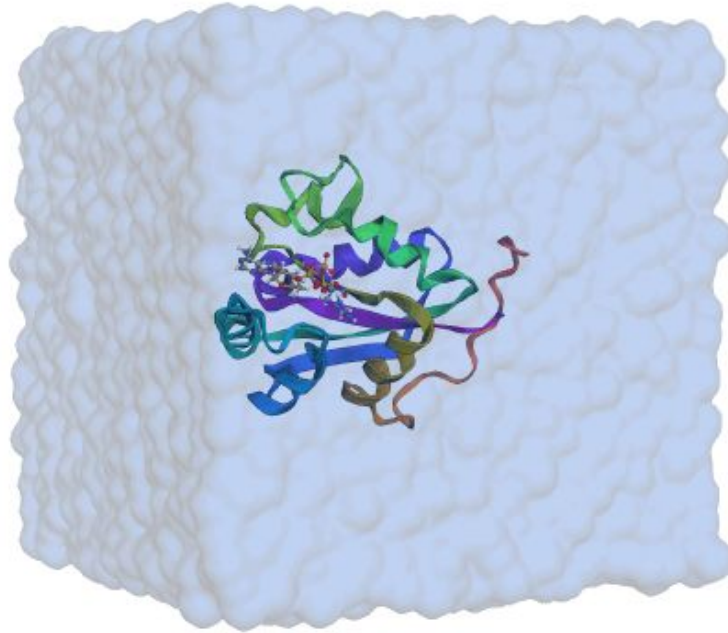
Meng Gao^{1,2,3}, Armin Shayesteh Zadeh^{1,2,3}, Aniruddha Seal¹, Siva Dasetty², Siddarth K. Achar², Misko Dzambaz¹, Benjamin K. Miller², Leif D. Jacobson¹, C. Lawrence Zitnick¹, Brandon M. Wood¹, Zachary W. Ulissi^{1,2}, Daniel S. Levine^{1,2}, Andrew L. Ferguson^{2,3}

¹FAIR at Meta, San Francisco, California, United States, ²Pritzker School of Molecular Engineering, University of Chicago, 5640 South Ellis Avenue, Chicago, Illinois 60637, United States, ³Department of Chemistry, University of Chicago, 5735 South Ellis Avenue, Chicago, Illinois 60637, United States

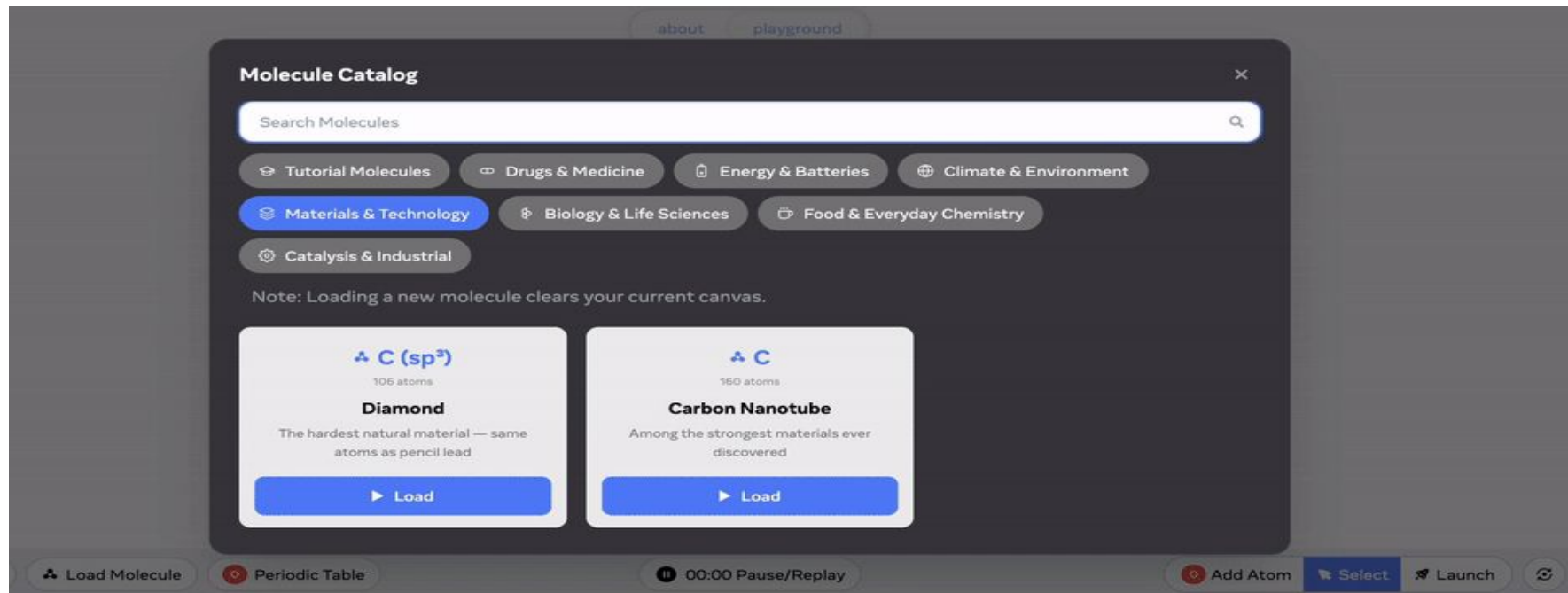
*These authors contributed equally and are listed alphabetically. *Co-corresponding authors

Standard deployment and coupling to rare-event enhanced sampling tools

Inside the enzyme – an atomistic view



UMA educational demo – explore molecules and materials



aidemos.atmeta.com/uma

Scan to open the interactive demo

THANK YOU!

Feedback, ideas, suggestions
welcome



From Neurons to Atoms: How PyTorch Powers Open Science at
Scale

