Scalable Training of Trustworthy and Energy-Efficient Predictive Graph Foundation Models for Atomistic Materials Modeling: A Case Study with HydraGNN

Massimiliano (Max) Lupo-Pasini

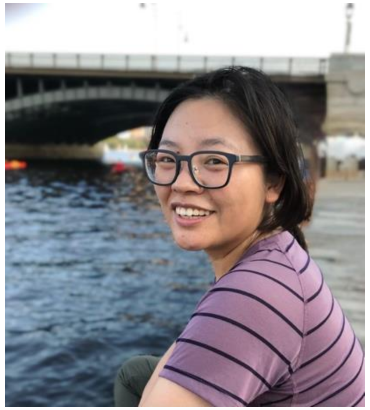
Oak Ridge National Laboratory

March 19, 2025

SOS27 Workshop, Engelberg, Switzerland

Team

Jong Youl Choi



Pei Zhang

Kshitij Mehta

David Rogers

Khaled Ibrahim
Lawrence Berkeley National Laboratory

Karl W. Schulz
AMD Research

Ashwin Aji
AMD Research

Jorda Polo
AMD Research

Prasanna Balaprakash

Driving motivation

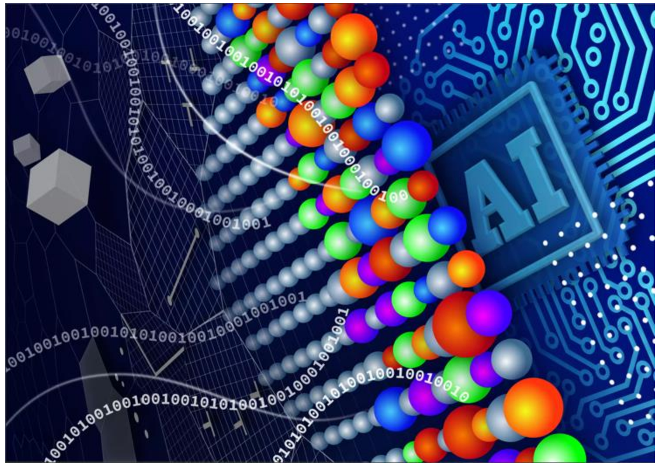
Design new materials with desired properties and functionalities from atomistic structures

Multiple applications:
- refractory high entropy alloys for fusion
- organic molecules with desired optoelectronic properties
- drug design
- energy innovation (batteries, superconductors, etc.)
- manufacturing

ORNL is the ideal environment to carry out this program due to its
- mission driven research portfolio
- unique facilities and recognized staff expertise
- access to scientific data
- collaborative environment

Current AI/ML approaches support the applications above, but require large amount of domain-specific training data.

Picture from https://www.printedelectronicsnow.com/contents/view_breaking-news/2018-12-24/ornl-new-composite-advances-lignin-as-renewable-3d-printing-material/



Picture from https://www.mpg.de/20096180/artificial-intelligence-in-material-design

Graph foundation models (GFMs)

In contrast, foundation models (FMs) are trained only once on generic data.

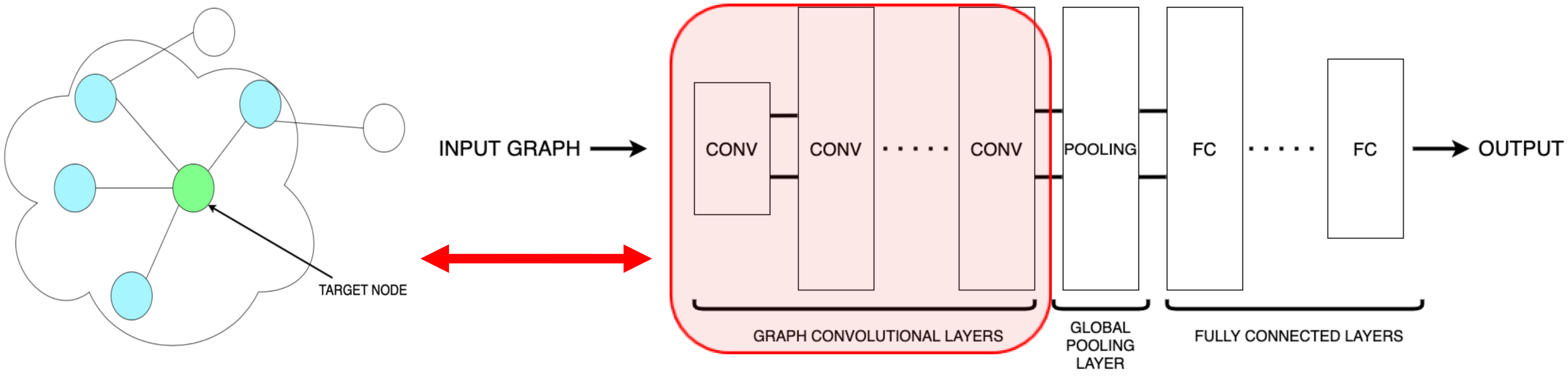
Once trained, FMs are fine-tuned on a wide variety of downstream tasks, with much less task-specific data.

- **Why GFMs ?**
  - Large Language Models (LLMs), originally developed for text applications, do not capture many aspects of materials' structure
  - The atomistic structure maps naturally onto the graph structure of the model
  - This facilitates developing physics-informed AI/ML methods
  - Graph neural networks (GNN) are the backbone of GFM architectures

Graph neural networks (GNNs)

The architecture of a GNN consists of:

1. a graph embedding layer
2. hidden graph layers to capture short range interactions between nodes in the graph
3. pooling layers interleaved with graph layers to synthetize information related to adjacent nodes via aggregation
4. fully connected (FC) dense layers at the end of the architecture to capture global features of the properties of interest



Convolutional operations aggregate information from neighboring nodes, thereby enabling transferability of local information to larger scales

OAK RIDGE National Laboratory

HydraGNN: a scalable GNN architecture for materials science applications

https://www.osti.gov/doecode/biblio/65891
https://github.com/ORNL/HydraGNN

Multi-task learning (MTL) from multiple source, heterogeneous, imbalanced data

Equivariance for efficient data processing and computational savings

Distributed data parallelism

HydraGNN: (i) supports continuing upgraded software; (ii) supports diverse scientific applications; and (iii) is portable across heterogeneous computing environments

Predicts simultaneously multiple quantities of interest

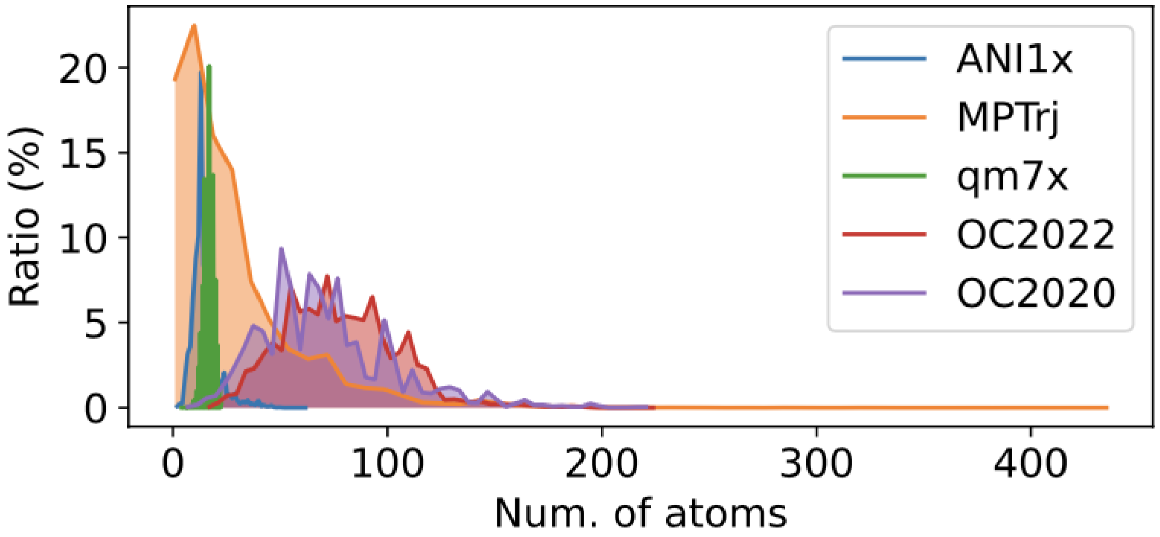
Stabilizes training avoids ill-conditioning and overfitting

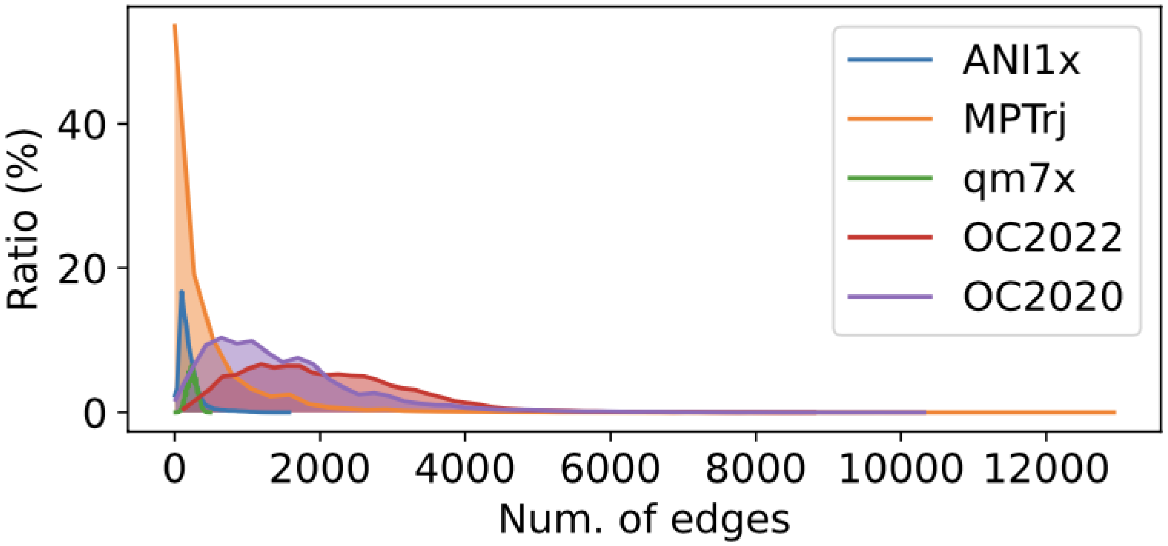
Ensures transferability

Efficient scaling

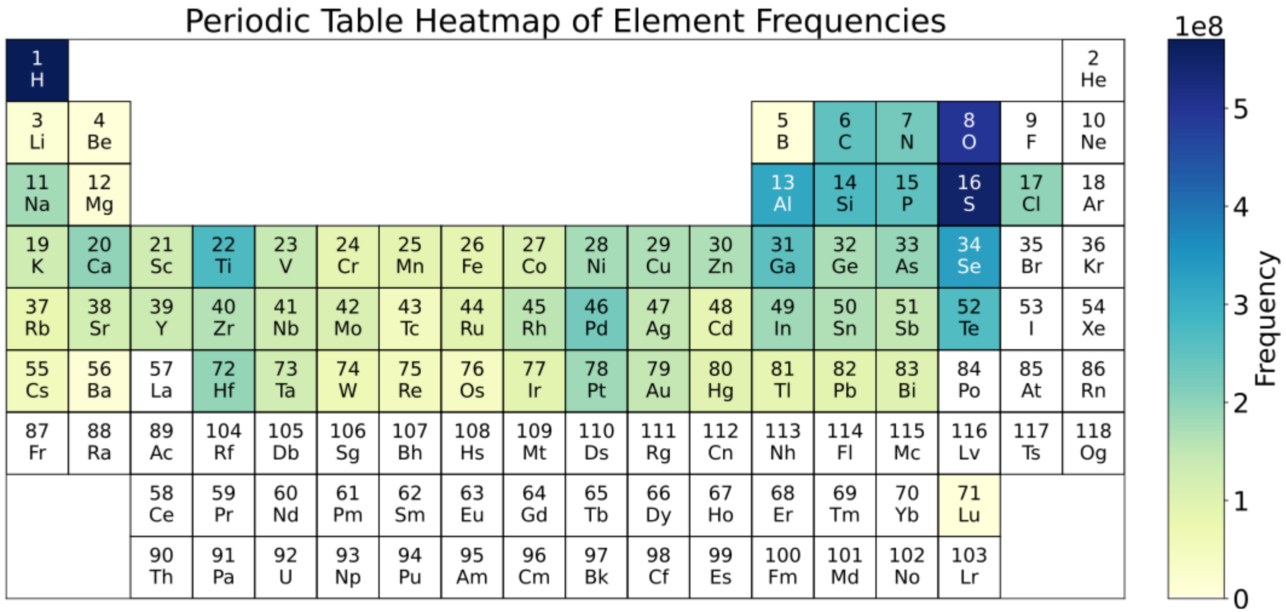
Pre-training of GFMs using HydraGNN

Aggregation of large volumes of open-source datasets





Dataset	Type of compounds	# data samples
Open Catalyst 2020	Alloy slabs with interacting catalysts on the surface	134,929,018
Open Catalyst 2022	Alloy slabs with interacting catalysts on the surface	8,847,031
Materials Project Trajectory	Bulk metals and alloys	1,580,395
ANI1x	Organic molecules	4,956,005
QM7x	Organic molecules	4,195,237
Total		**154,507,686**



Oak Ridge National Laboratory

Data cleaning

Dataset	Number of data samples removed
ANI1x [67]	0
QM7-X [68]	0
OC2020 [43]	1
OC2022 [44]	12,270
MPTrj [41]	151
Total	12,421

We removed data-samples with L2-norm of the force-tensor above 100 eV/angstrom

(eV)

(eV/angstrom)

Data alignment

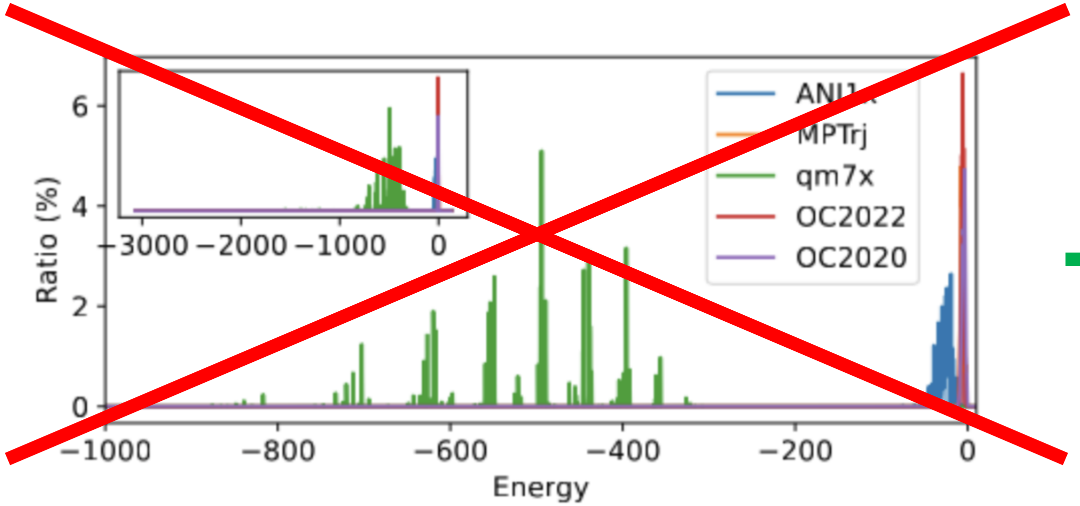
We **re-aligned** multi-source multi-fidelity data to transform the energy of each atomistic structure during pre-processing by subtracting a linear regression term. For each dataset, the linear regression term was calculated by solving the following least-squares problem:

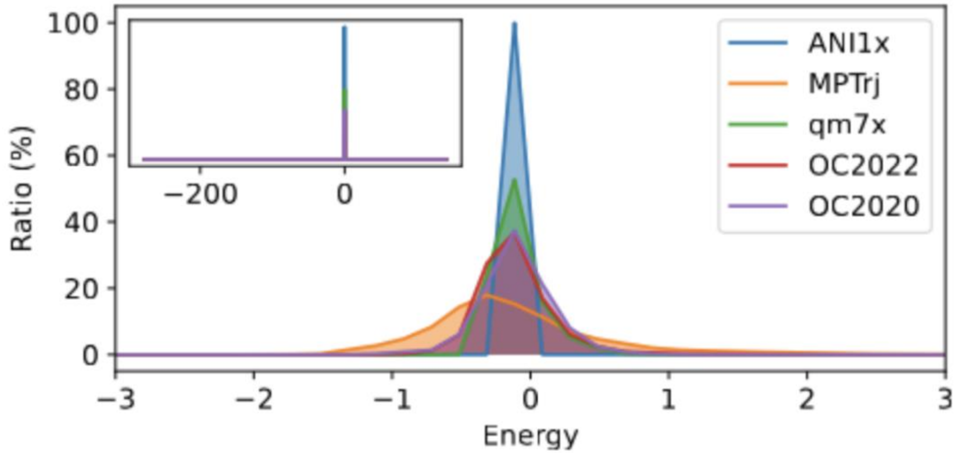
$$\underset{C_1,\ldots,C_{118}}{\arg\min} \sum_{i=1}^{N_{data}} \left(e_0^i - \sum_{Z=1}^{118} C_Z n_Z^i \right)^2$$

e_0^i = reference energy for the atomistic structure ***i***

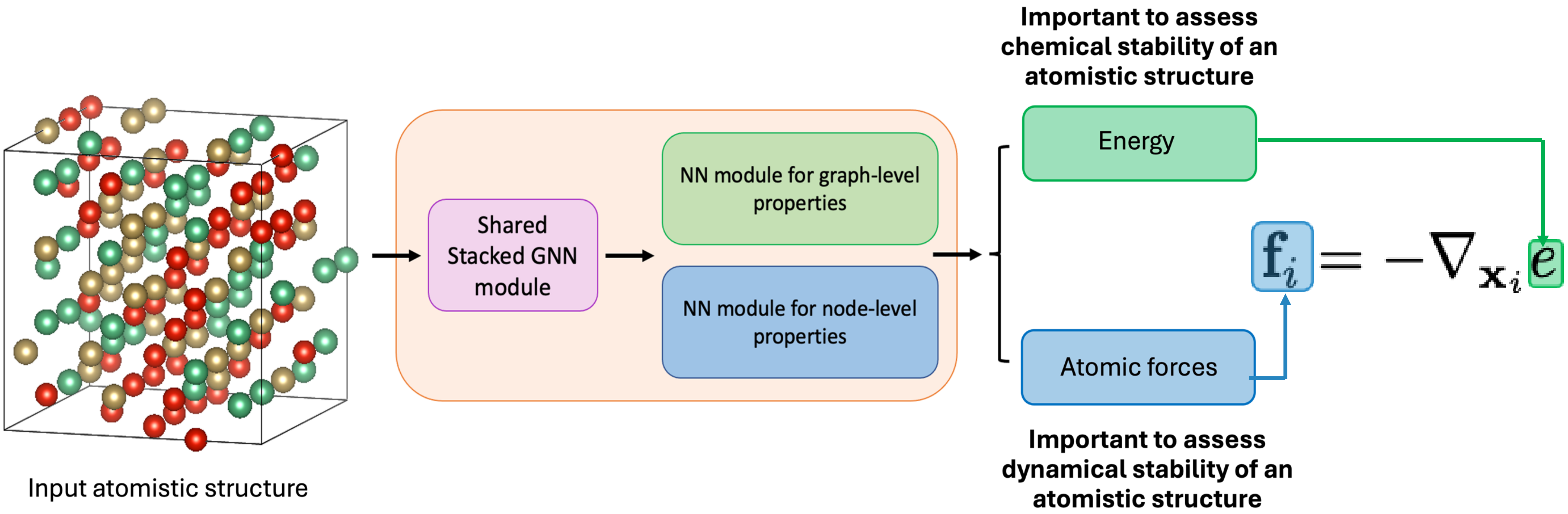
n_Z^i = number of atoms for element number ***Z*** for belonging to the atomistic structure ***i***

C_Z = regression coefficients that needs to be computed





Multi-tasking HydraGNN architecture

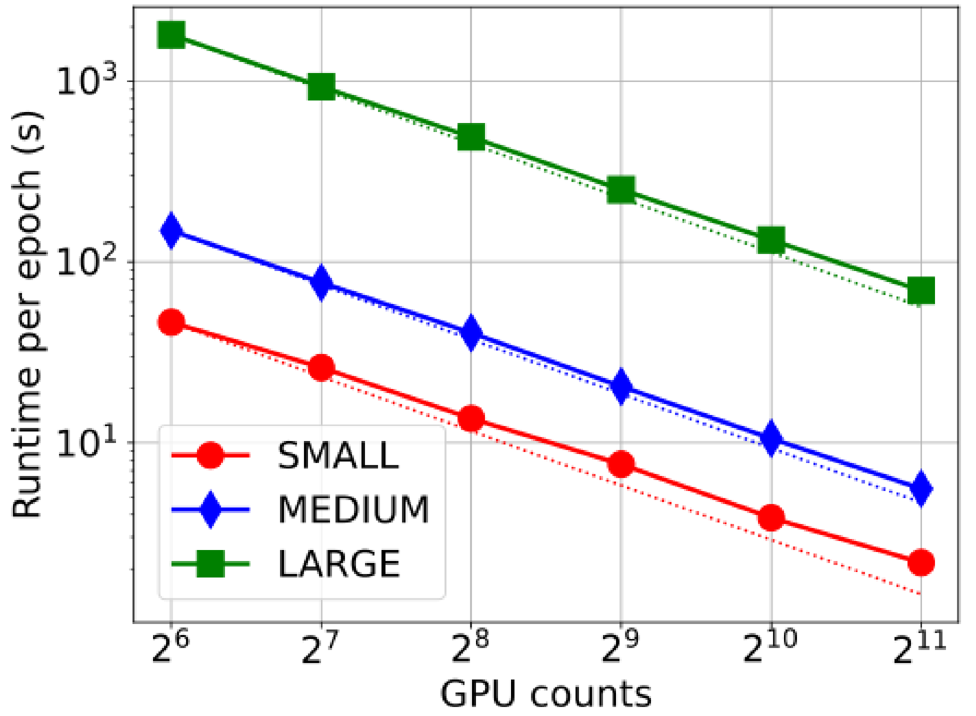


Strong scaling performance

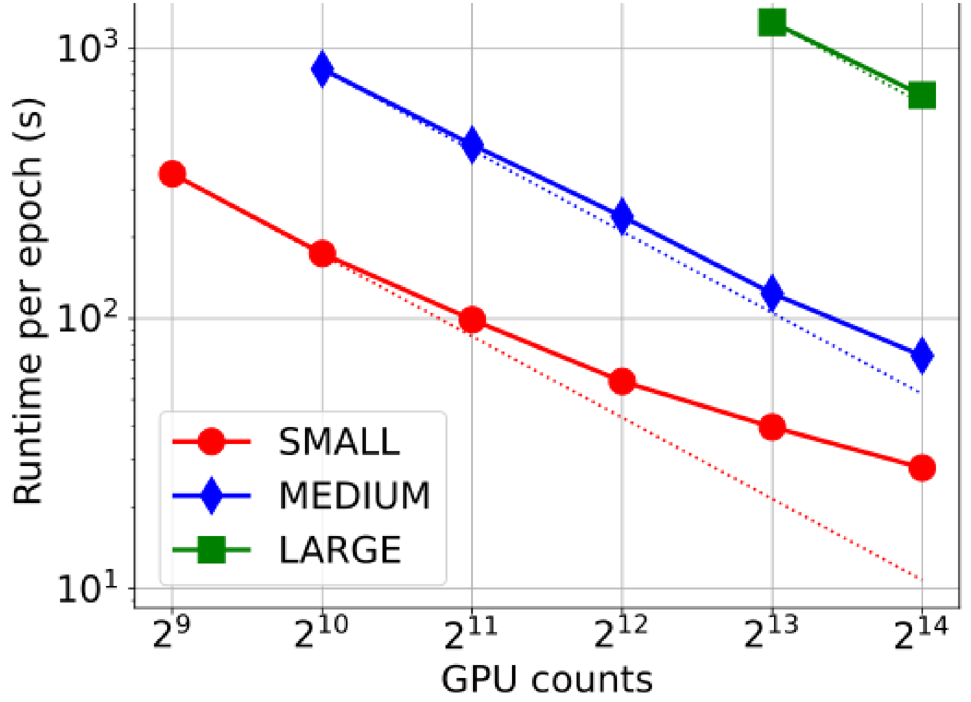
Model size	SMALL	MEDIUM	LARGE
Type of MPNN layer	EGNN	EGNN	EGNN
# MPNN layers	3	6	6
# neurons in MPNN layers	50	500	2,000
# FC layers	2	2	3
# neurons in FC layers	50	1,000	1,000
Number of parameters	**58,404**	**14,539,004**	**163,129,004**

Top: GNN model sizes used for strong scaling tests on NERSC-Perlmutter and OLCF-Frontier.

Perlmutter



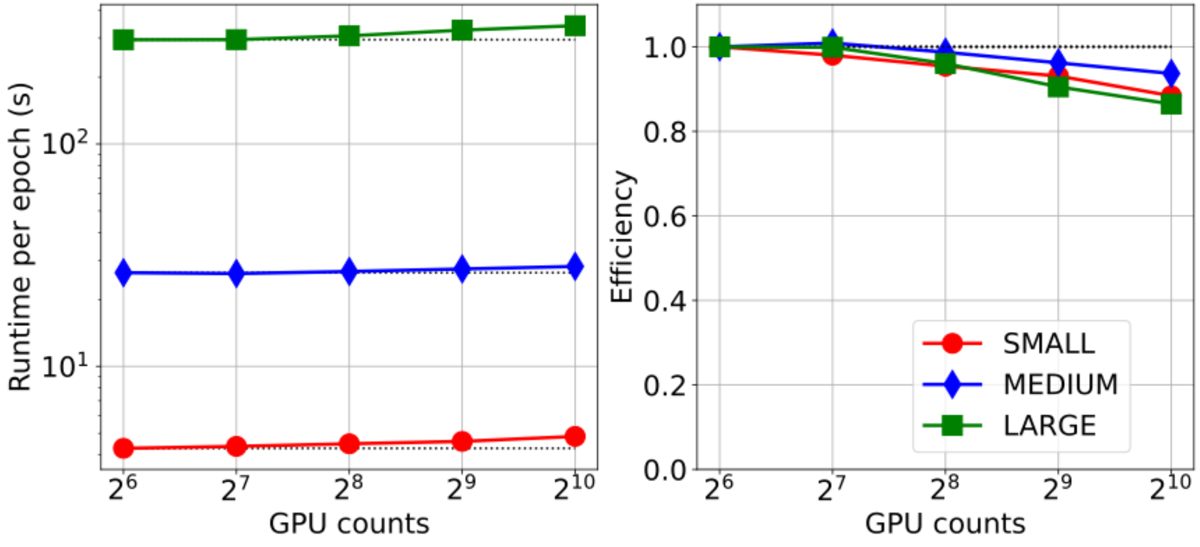
Frontier



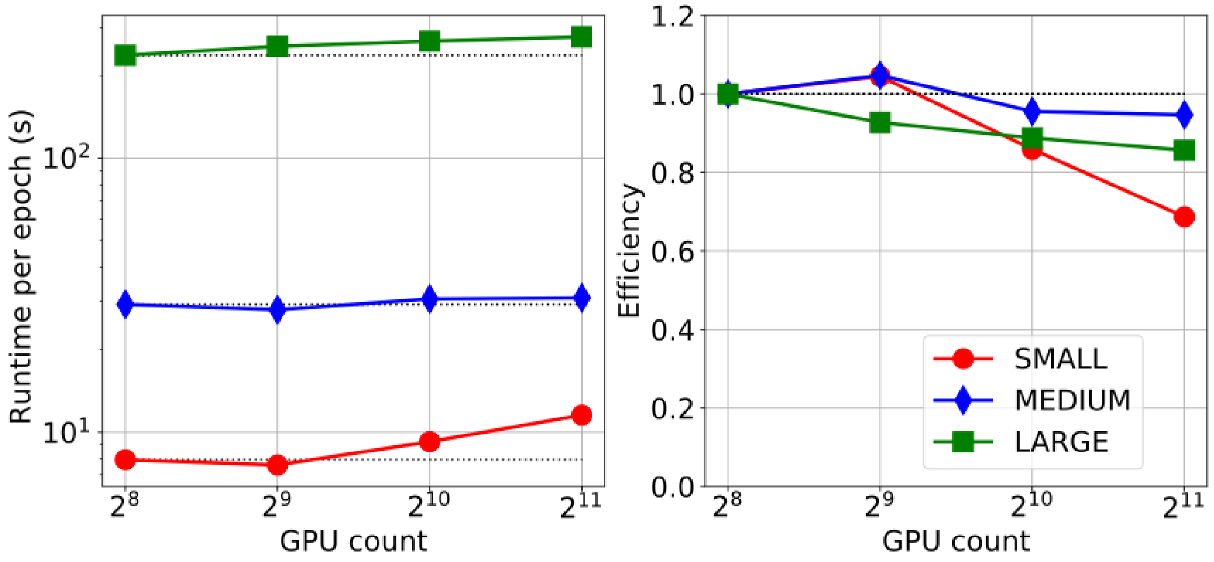
Weak scaling performance

Model size	SMALL	MEDIUM	LARGE
Type of MPNN layer	EGNN	EGNN	EGNN
# MPNN layers	3	6	6
# neurons in MPNN layers	50	500	2,000
# FC layers	2	2	3
# neurons in FC layers	50	1,000	1,000
Number of parameters	**58,404**	**14,539,004**	**163,129,004**

Top: GNN model sizes used for strong scaling and weak scaling tests on NERSC-Perlmutter and OLCF-Frontier.



Perlmutter



Frontier

OAK RIDGE National Laboratory

Hyperparameter optimization (HPO)

Challenges:

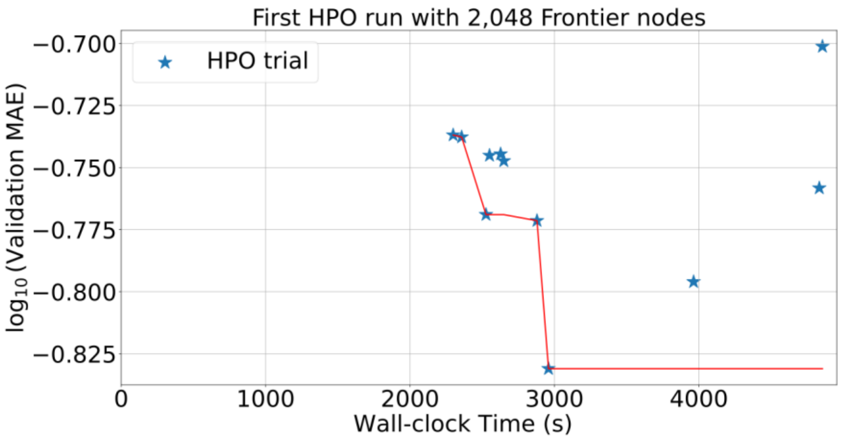
- The hyperparameter space is high-dimensional, thus making its exploration combinatorically complex
- The GFM training for each HPO trial needs to scale on 128 Frontier nodes to effectively process large volumes of data using DDP

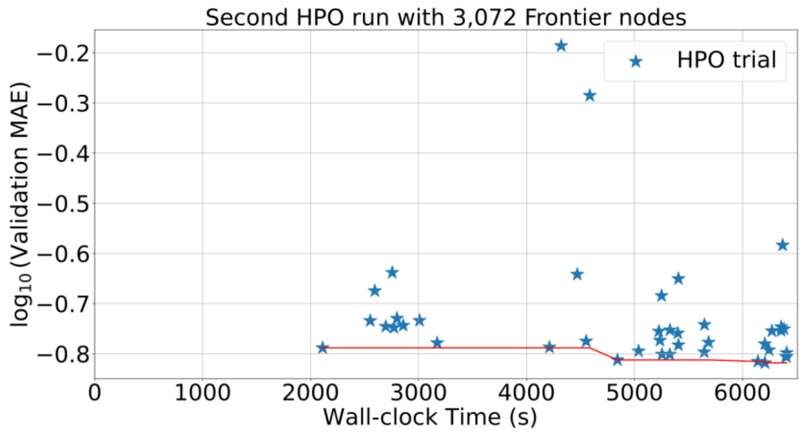
→ Need for massive computational resources (and energy)

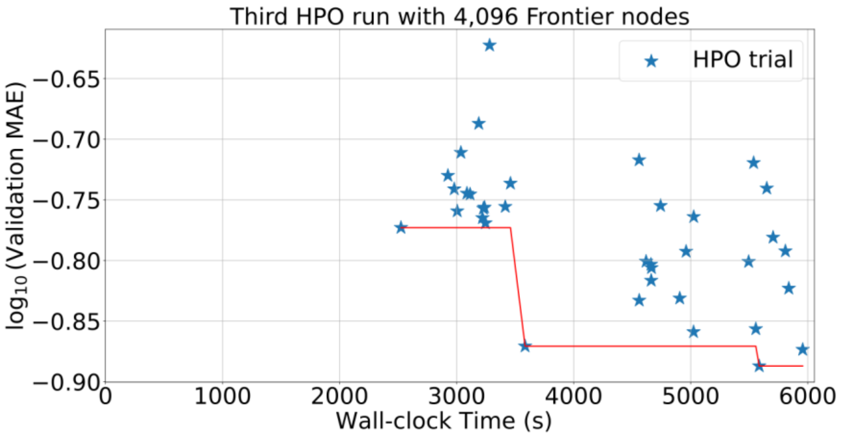
Proposed approach:

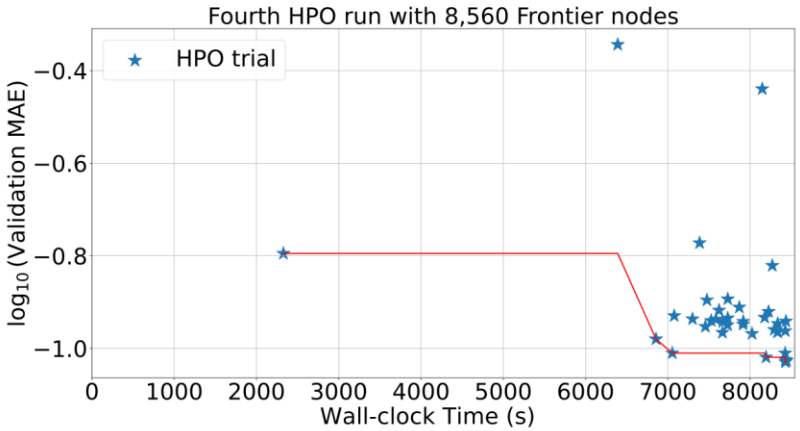
- Progressively ramp-up the scale of HPO with a funneling approach, whereby preliminary HPO runs at smaller scale instructing successive HPO runs at larger scale
- Select GFMs with optimal balance between accuracy vs. energy-savings
- Complete training till convergence is reached only for selected GFMs

HPO runs with increasing scale up to 8,560 Frontier nodes









Progressive decrease of the validation loss function through consecutive HPO runs

‘Omnistat’ for collection of key telemetry from AMD GPUs

Github repository: https://github.com/AMDResearch/omnistat

Omnistat provides a set of utilities to aggregate scale-out system metrics via low-overhead sampling across all hosts in a cluster or, alternatively on a subset of hosts associated with a specific user job.

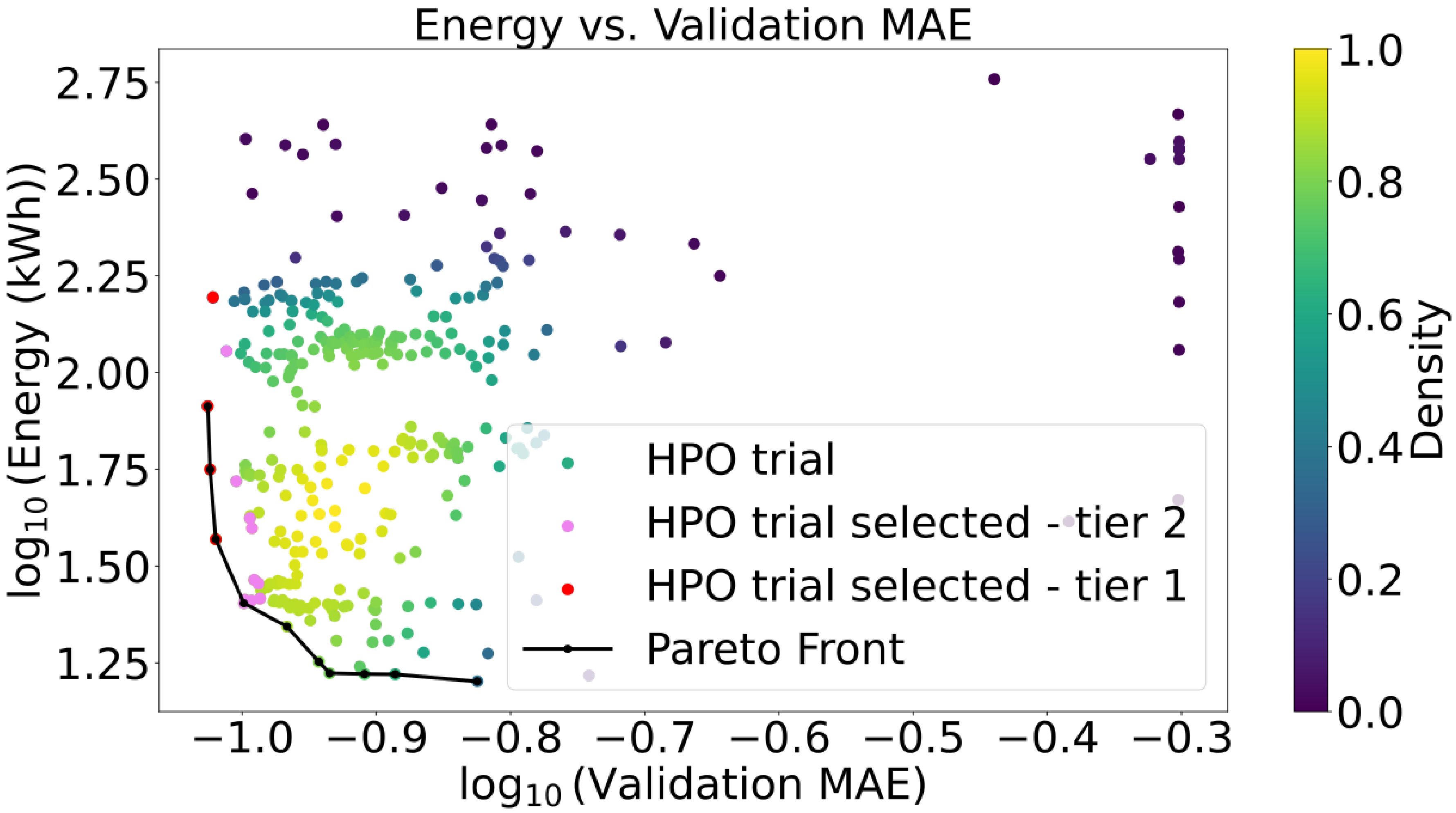
Relevant target metrics include:

- GPU utilization (occupancy)
- High-bandwidth memory (HBM) usage
- GPU power
- GPU temperature
- GPU clock frequency (Mhz)
- GPU memory clock frequency (Mhz)
- GPU throttling events

Selection of HPO trials with optimal trade-off between accuracy and energy-savings

Tier 1:
4 GFMs with clear advantage in accuracy

Tier 2:
11 GFMs with best trade-off between accuracy and energy savings

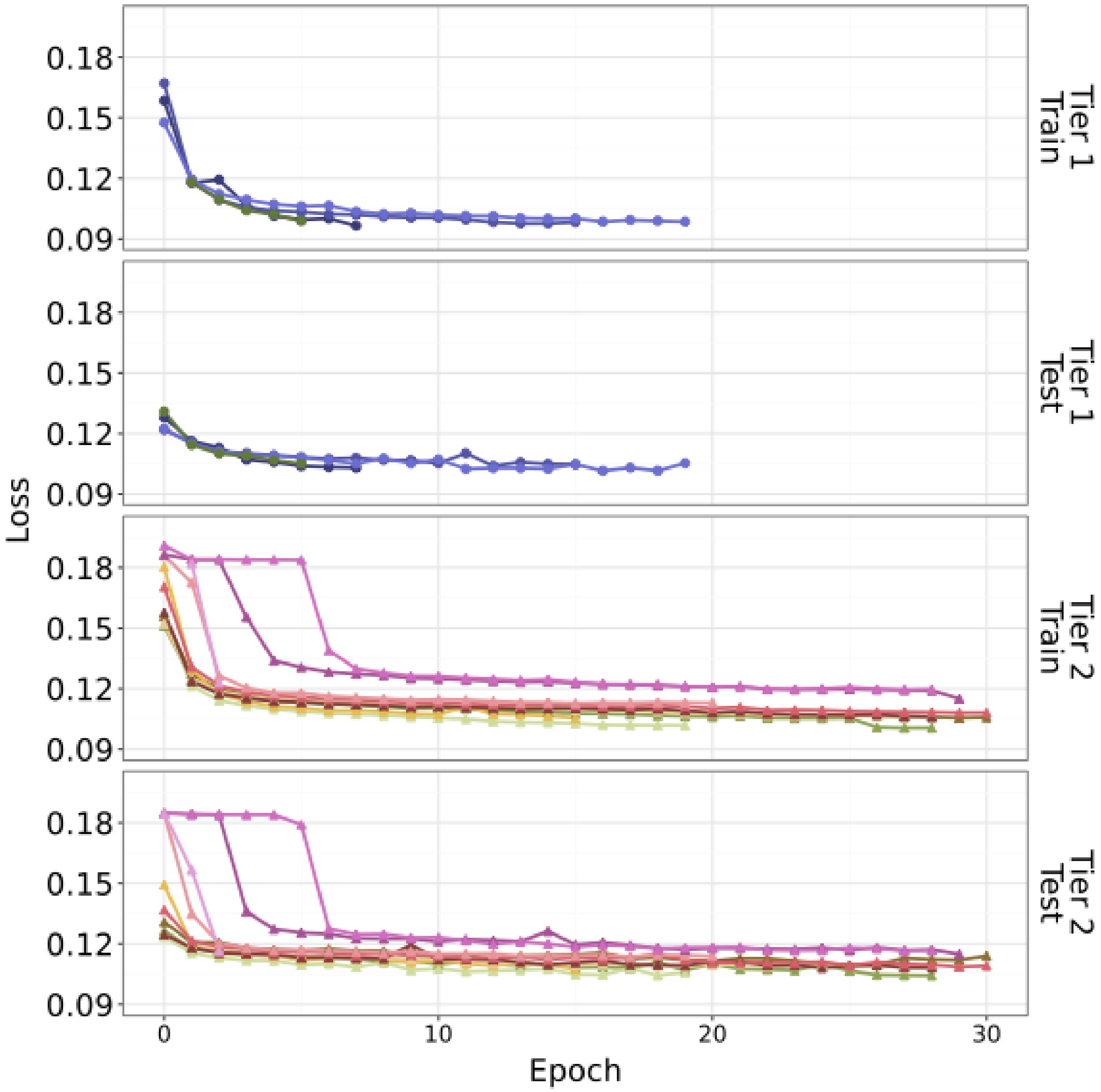


Full training of selected HPO trials

Best HPO trials are trained till convergence using checkpoint restart strategies to facilitate resubmission of job on OLCF-Frontier.

Early stopping is applied to force interruption when validation loss function does not significantly improve across 10 consecutive epochs.

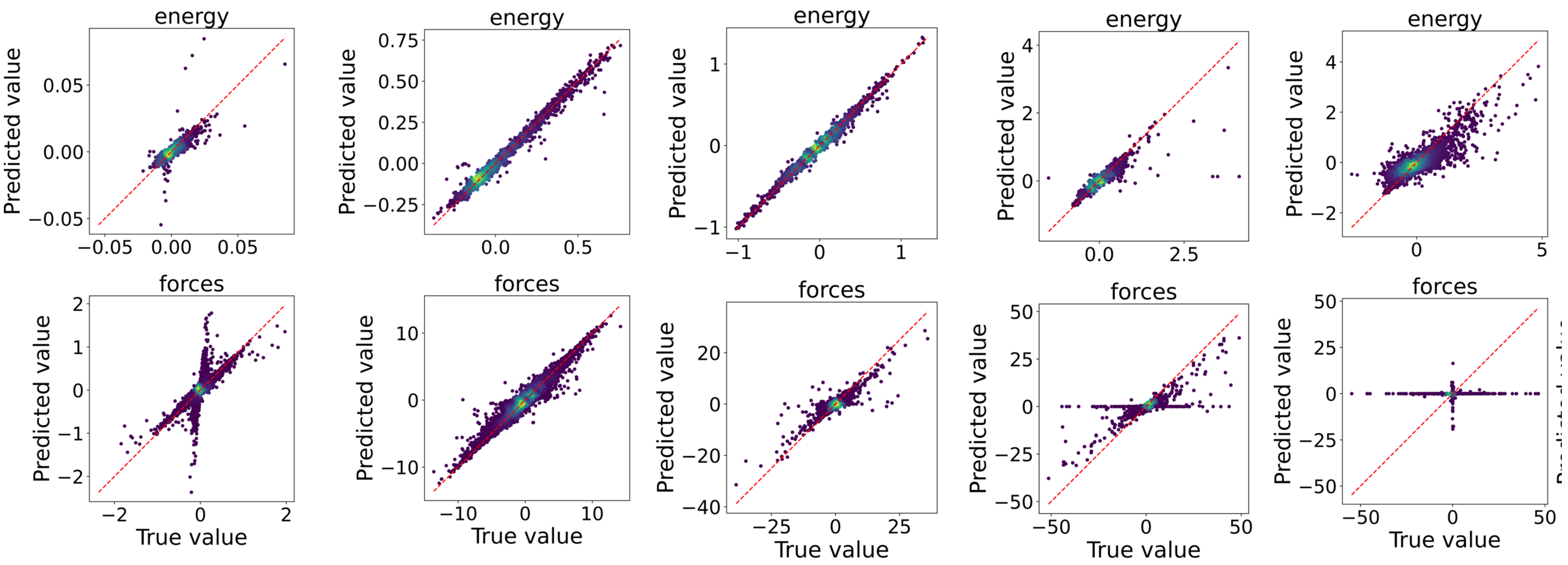
Training of each GFM took approximately 10 wall-clock hours using 128 nodes for DDP



OAK RIDGE National Laboratory

Ensemble learning performance

ANI1-X | QM7-X | OC2020 | OC2022 | Materials Project

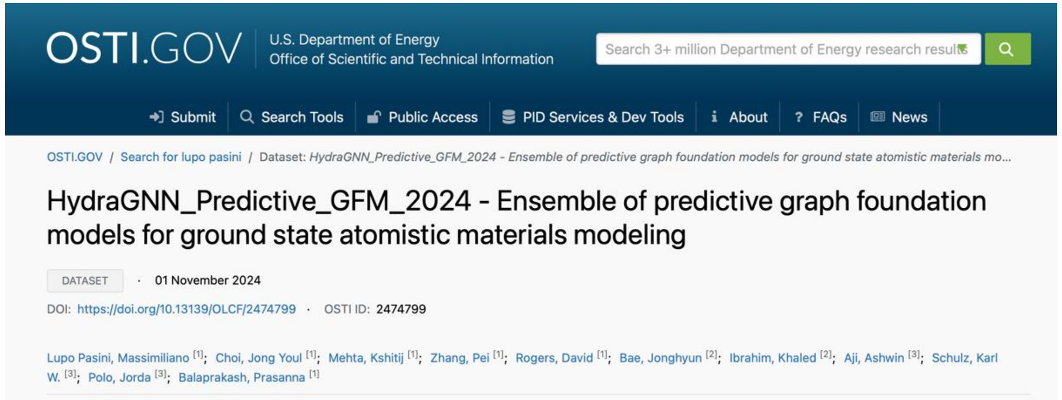


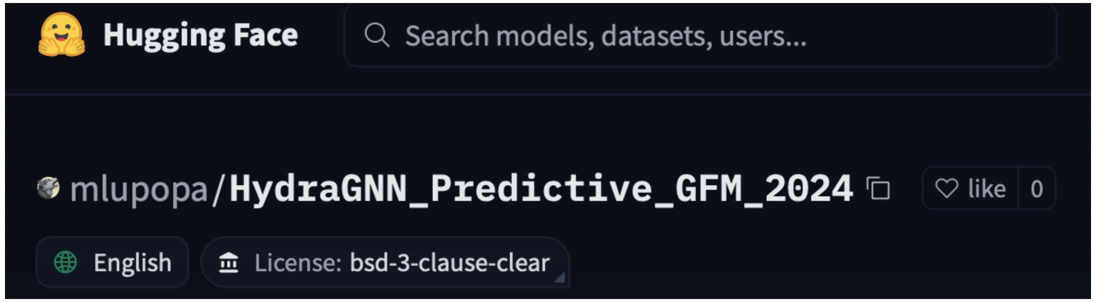
Publication and release of GFM parameters with examples

Publication:
M. Lupo Pasini, J. Y. Choi, K. Mehta, P. Zhang, D. Rogers, J. Bae, K. Ibrahim, A. Aji, K. W. Schulz, J. Polo, and P. Balaprakas. *Scalable training of trustworthy and energy-efficient predictive graph foundation models for atomistic materials modeling: a case study with HydraGNN*.
J Supercomput 81, 618 (2025). https://doi.org/10.1007/s11227-025-07029-9

Models and pre-processed datasets released on OSTI.GOV:
M. Lupo Pasini, J. Y. Choi, K. Mehta, P. Zhang, D. Rogers, J. Bae, K. Ibrahim, A. Aji, K. W. Schulz, J. Polo, and P. Balaprakash (2024). *HydraGNN_Predictive_GFM_2024 - Ensemble of predictive graph foundation models for ground state atomistic materials modeling*.
https://doi.org/10.13139/OLCF/2474799

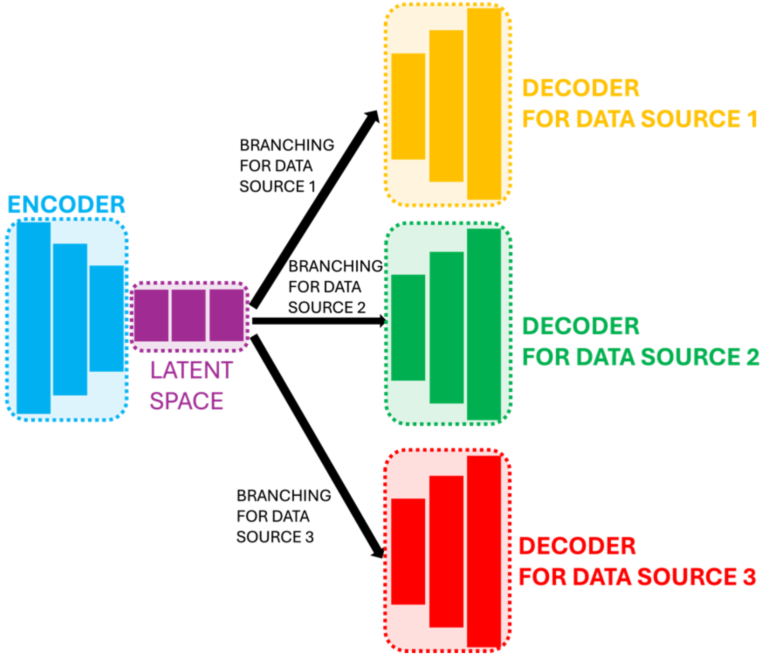
Models with example scripts to load them available on Hugging Face:
https://huggingface.co/mlupopa/HydraGNN_Predictive_GFM_2024

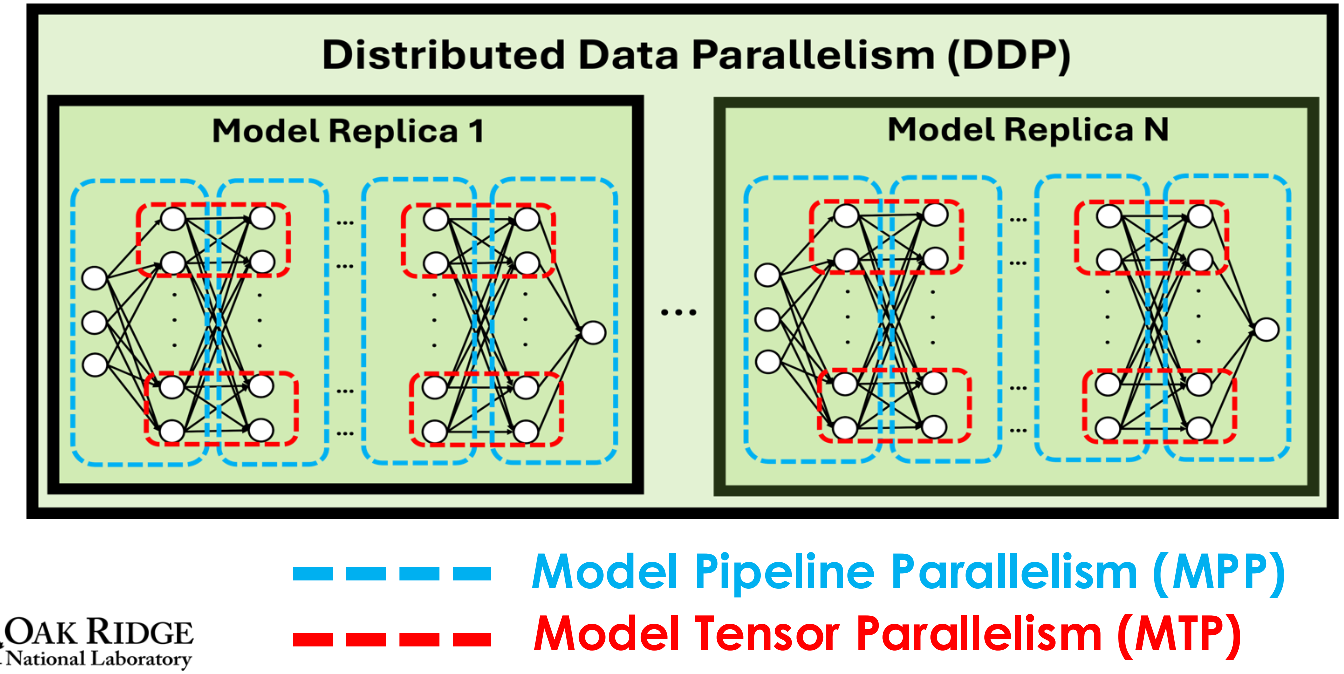




On-going and future work

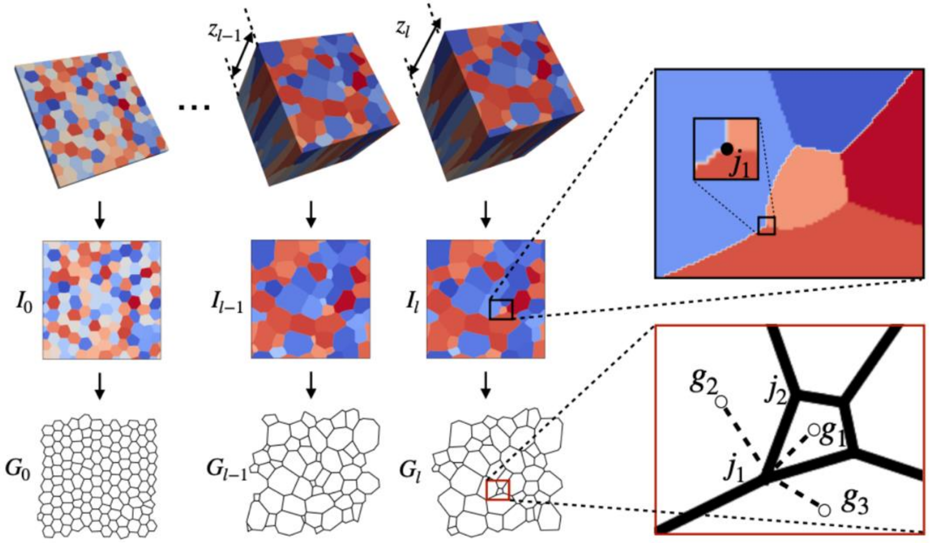
- Integrate DDP, model parallelism (MP)
- Generalize MTL for stable and energy-efficient training
- Develop equivariant generative diffusion models with masking techniques
- Apply HydraGNN to modeling evolution of grain structure at mesoscopic scale





Mesoscale

Nodes = Voronoi centers
Edges = connection between Voronoi centers



OAK RIDGE National Laboratory

Acknowledgments

This research is partially supported by:

- The Artificial Intelligence Initiative as part of the Laboratory Directed Research and Development (LDRD) Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the US Department of Energy under contract DE-AC05-00OR22725.

- The SciDAC Institute for Computer Science, Data, and Artificial Intelligence (RAPIDS), Lawrence Berkeley National Laboratory, which is operated by the University of California for the U.S. Department of Energy under contract DE-AC02-05CH11231.

- The US-DOE Advanced Scientific Computing Research (ASCR) under contract DE-SC0023490.

This research used resources of the Oak Ridge Leadership Computing Facility and of the Edge Computing program at the Oak Ridge National Laboratory. Computer time was provided by the INCITE program using the OLCF award CPH161 (AY2024), and OLCF Director's Discretion Project program using the OLCF awards MAT250 (AY2022) and LRN026 (AY2023).

This research also used resources of the National Energy Research Scientific Computing Center (NERSC), which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725 and No. DE-AC02-05CH11231, using NERSC awards ASCR-ERCAP0022058 (AY2022), ASCR-ERCAP0025216 (AY2023) and ASCR-ERCAP0027259 (AY2024).

Thank you!

Questions?

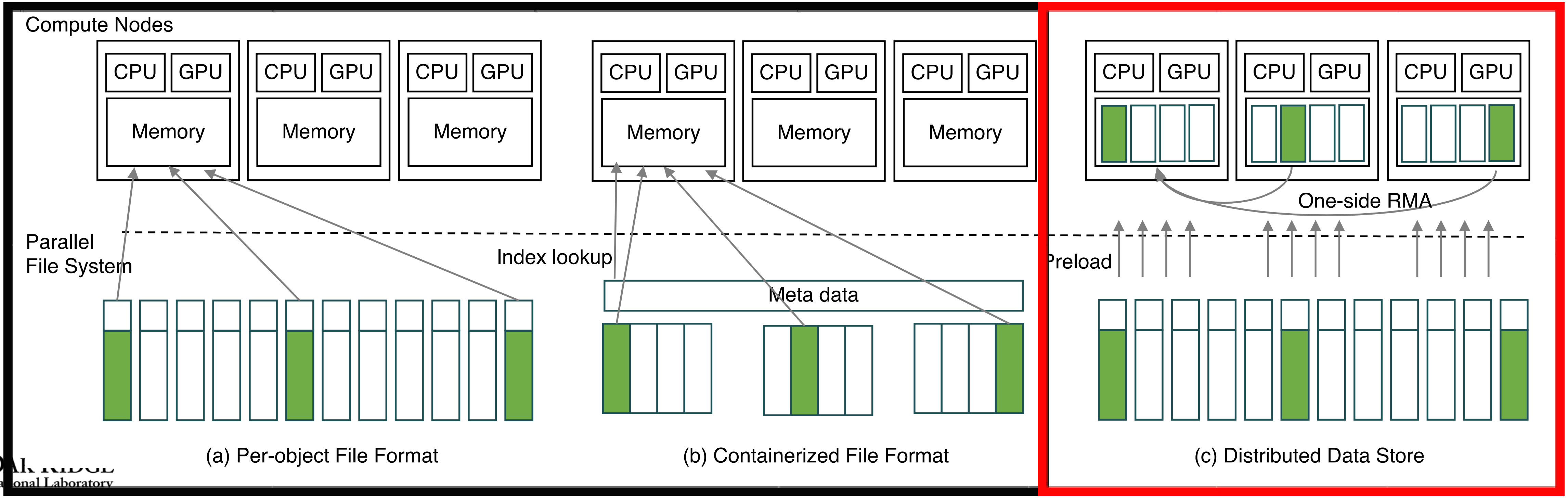
HydraGNN: scalable training with distributed data parallelism (DDP)

Traditional form of DDP :

- Move all data onto the memory of one compute unit
- Periodically pull data from the parallel file system (PFS

Distributed Data Store (DDStore) library partitions data in chunks and moves it from the PFS to the compute memory of each node. This:

- **Helps scaling DDP for data that CANNOT be stored within the memory of one compute unit**
- **Avoids frequent communications with the PFS**

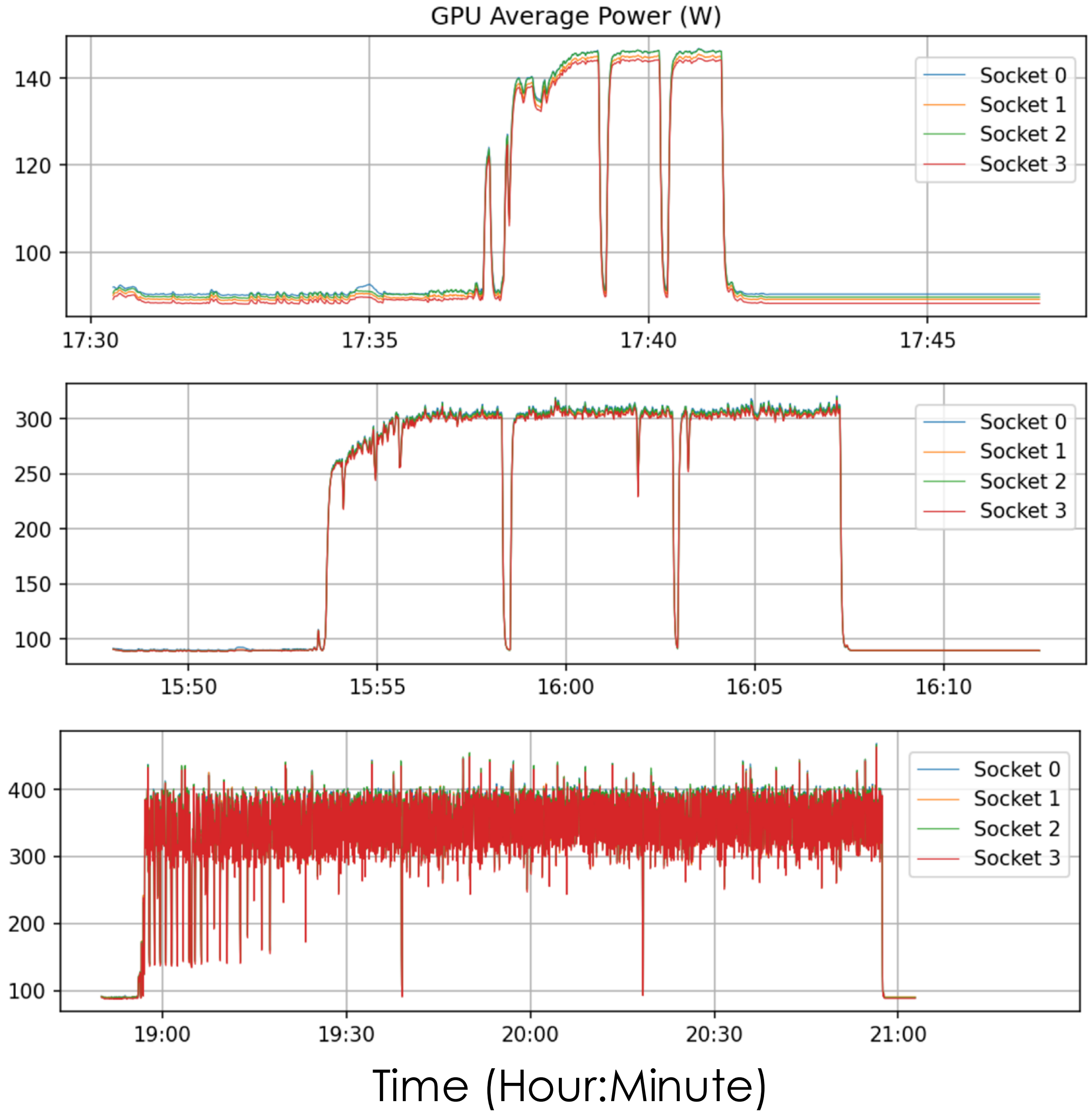


GPU power measurement for models of different size

Model size	SMALL	MEDIUM	LARGE
Type of MPNN layer	EGNN	EGNN	EGNN
# MPNN layers	3	6	6
# neurons in MPNN layers	50	500	2,000
# FC layers	2	2	3
# neurons in FC layers	50	1,000	1,000
Number of parameters	**58,404**	**14,539,004**	**163,129,004**

Top: GNN model sizes used for strong and weak scaling tests on NERSC-Perlmutter and OLCF-Frontier.

Right: GPU Energy use over time for three models – SMALL (top), MEDIUM (middle), and LARGE (bottom). Each line represents one AMD Instinct MI250x.



Synergy between experimental and computational DOE user facilities

Manufacturing Demonstration Facility

Spallation Neutron Source

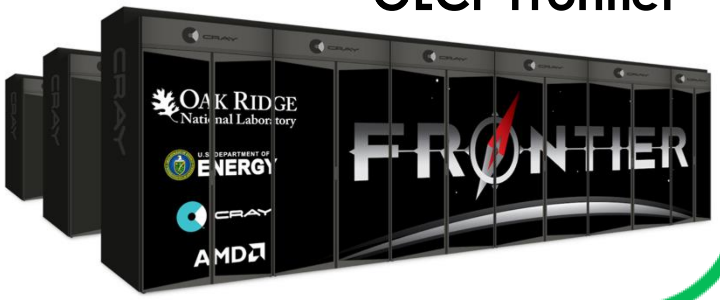
Center for Nanophase Material Sciences

Experimental data to train AI models

Validation of AI outcomes with experimental data

NERSC Perlmutter

OLCF Frontier

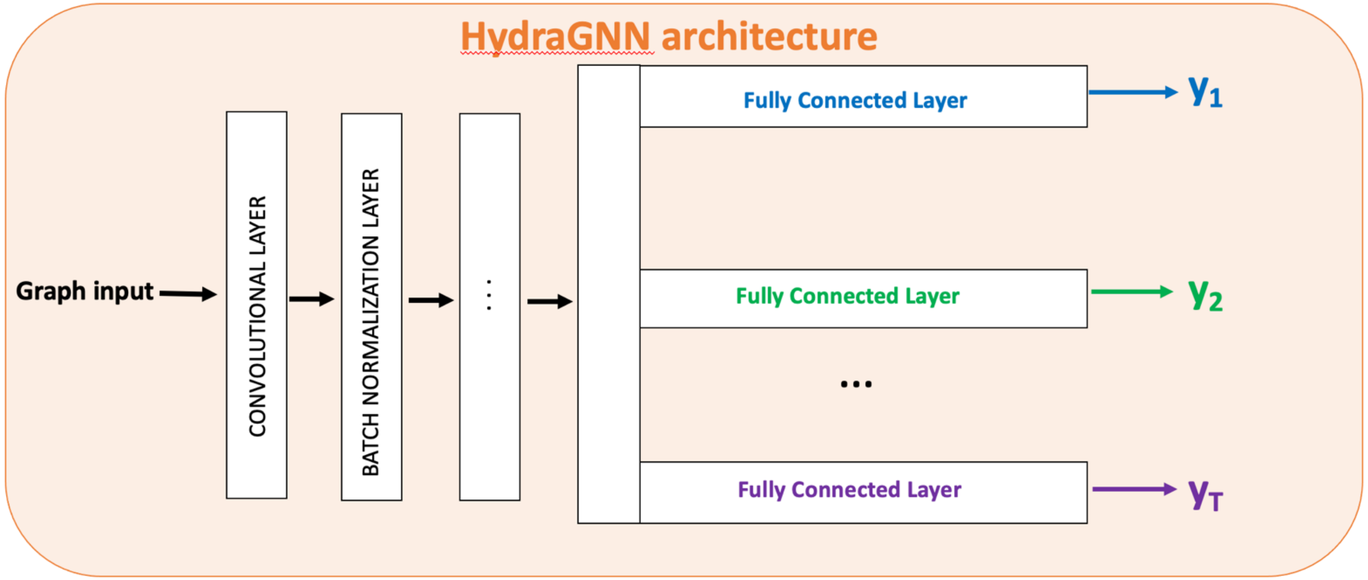


HydraGNN: multi-task learning

Multi-Task Learning stabilizes predictions of multiple properties
Each property operates as a mutual regularizer on the other properties

Quantities simultaneously predicted:

- **Property y_1**
- **Property y_2**
- …
- **Property y_T**



$\mathbf{w}$ = parameters of the neural network to optimize during training

$$\underset{\mathbf{w}}{\operatorname{argmin}} \underbrace{\boxed{\|\mathbf{y}_{\text{predict},1}(\mathbf{w}) - \mathbf{y}_1\|_2^2} + \boxed{\|\mathbf{y}_{\text{predict},2}(\mathbf{w}) - \mathbf{y}_2\|_2^2} + \ldots + \boxed{\|\mathbf{y}_{\text{predict},T}(\mathbf{w}) - \mathbf{y}_T\|_2^2}}_{\text{Global Multi-Task Training Loss Function}}$$

HydraGNN: equivariance

Equivariance is the property that, under Euclidean transformations, maintains consistency between the geometric structure and the physical properties associated with it.
This property is stronger than regular invariance that maintains only geometric properties.

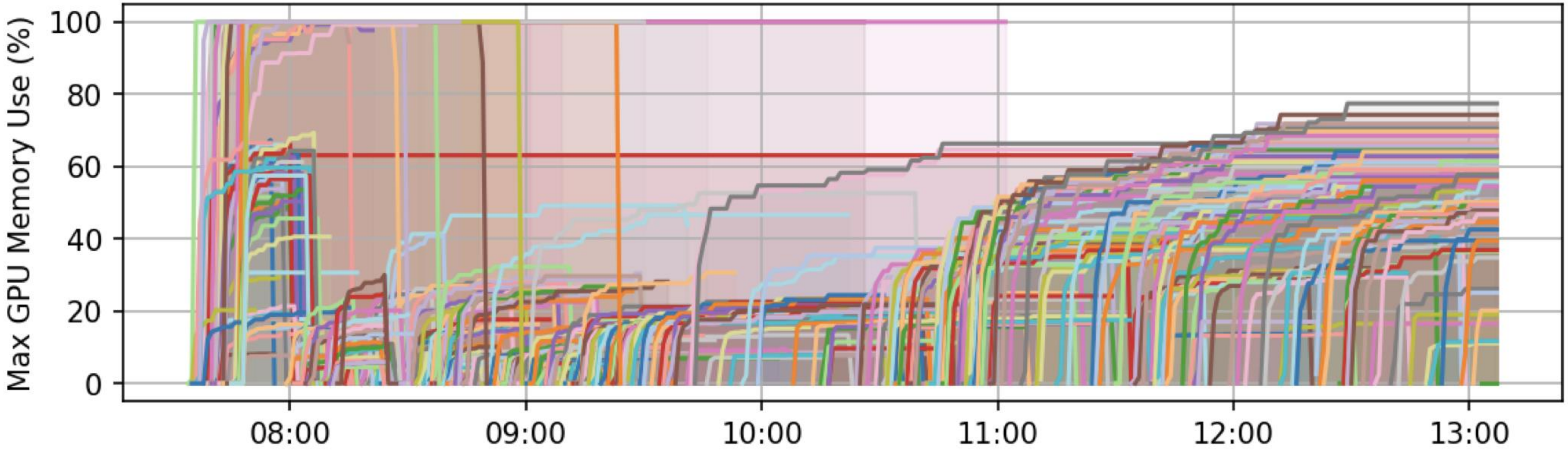
Equivariance collapses the whole class of structurally and functionally equivalent compounds into just one representative.

Implementing equivariance in the message passing layers acts as an **inductive bias.**
It eliminates data redundancy and reduces the computational cost to reach the desired accuracy. ***This is expected also to reduce energy consumption.***

Examples of invariant material properties:
HOMO-LUMO gap, **free energy**, vibrational spectrum, electronic excitation spectrum

Examples of equivariant material properties:
Electron charge density, **atomic forces**, magnetic moment

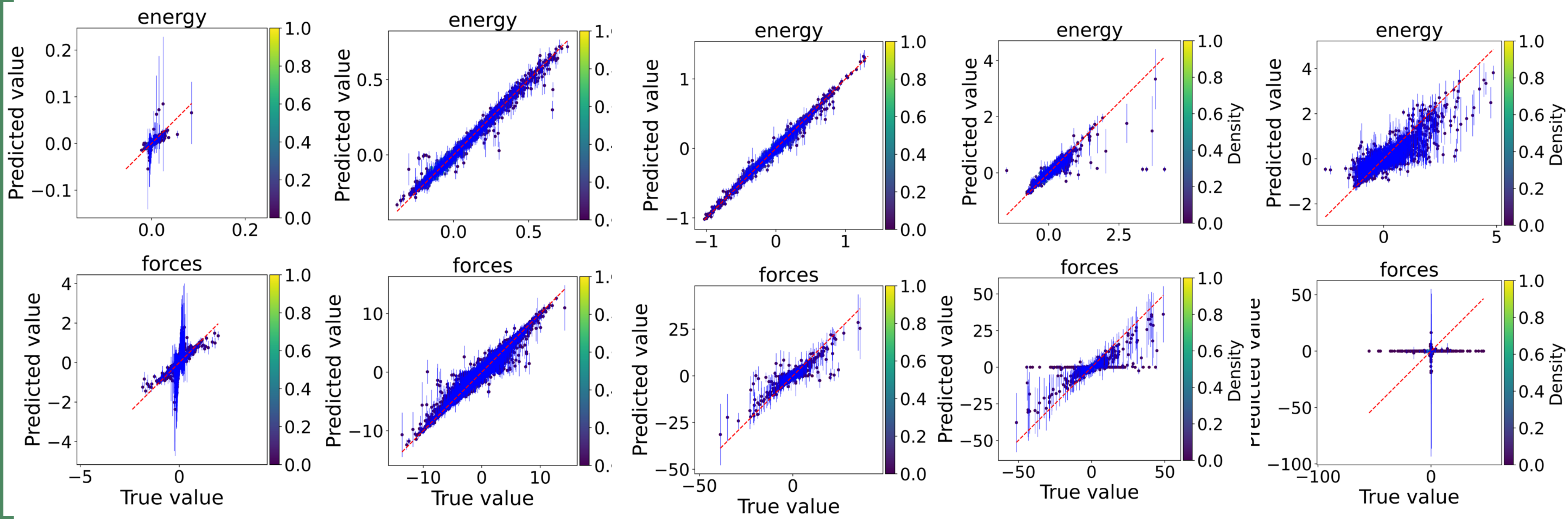
GPU memory utilization and power measurement



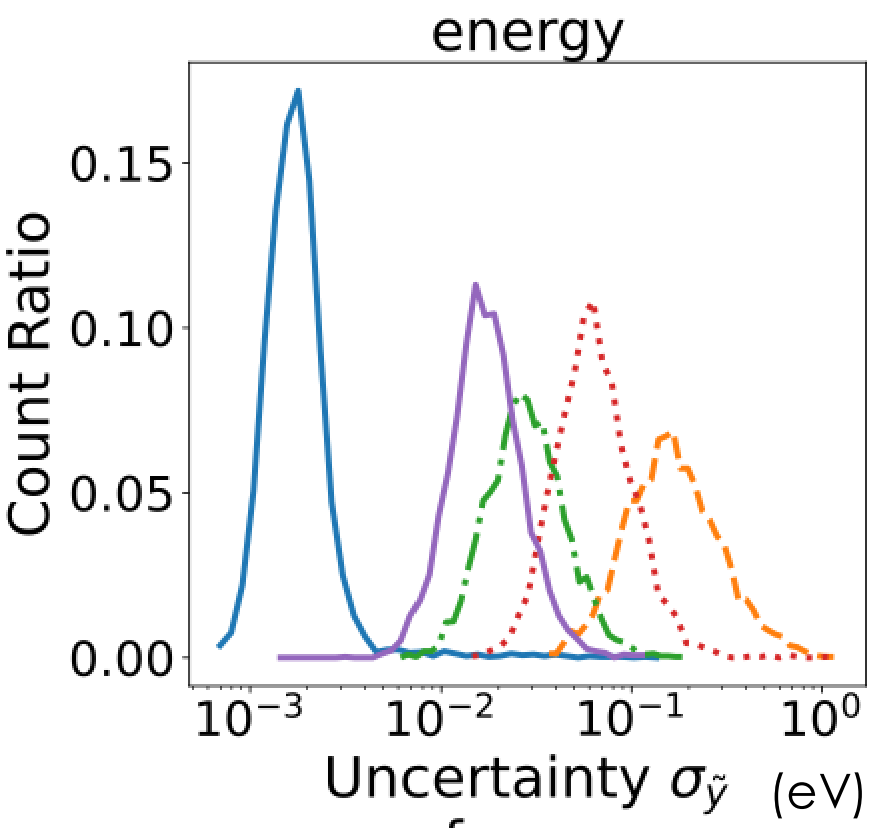
Max GPU HBM memory consumption traces sampled via Omnistat telemetry harness during final HPO exercise using 8,560 Frontier nodes (68,480 GCDs) executed on OLCF-Frontier.

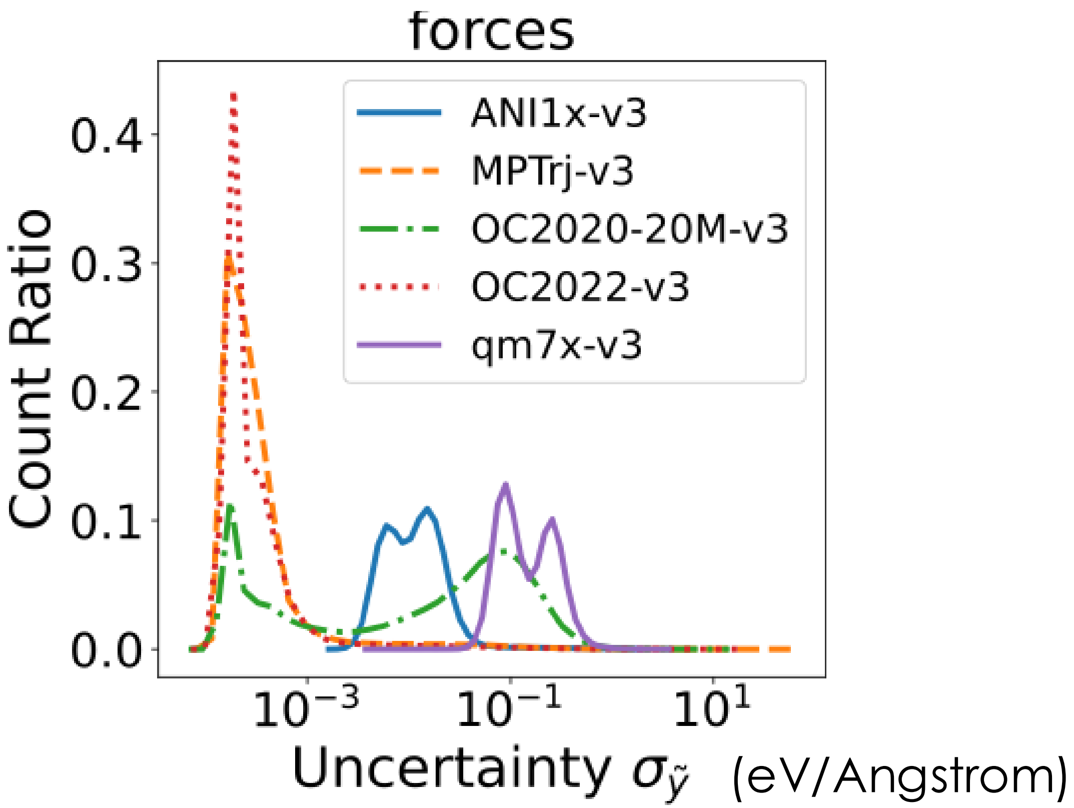
Ensemble uncertainty quantification

ANI1-X | **QM7-X** | **OC2020** | **OC2022** | **Materials Project**



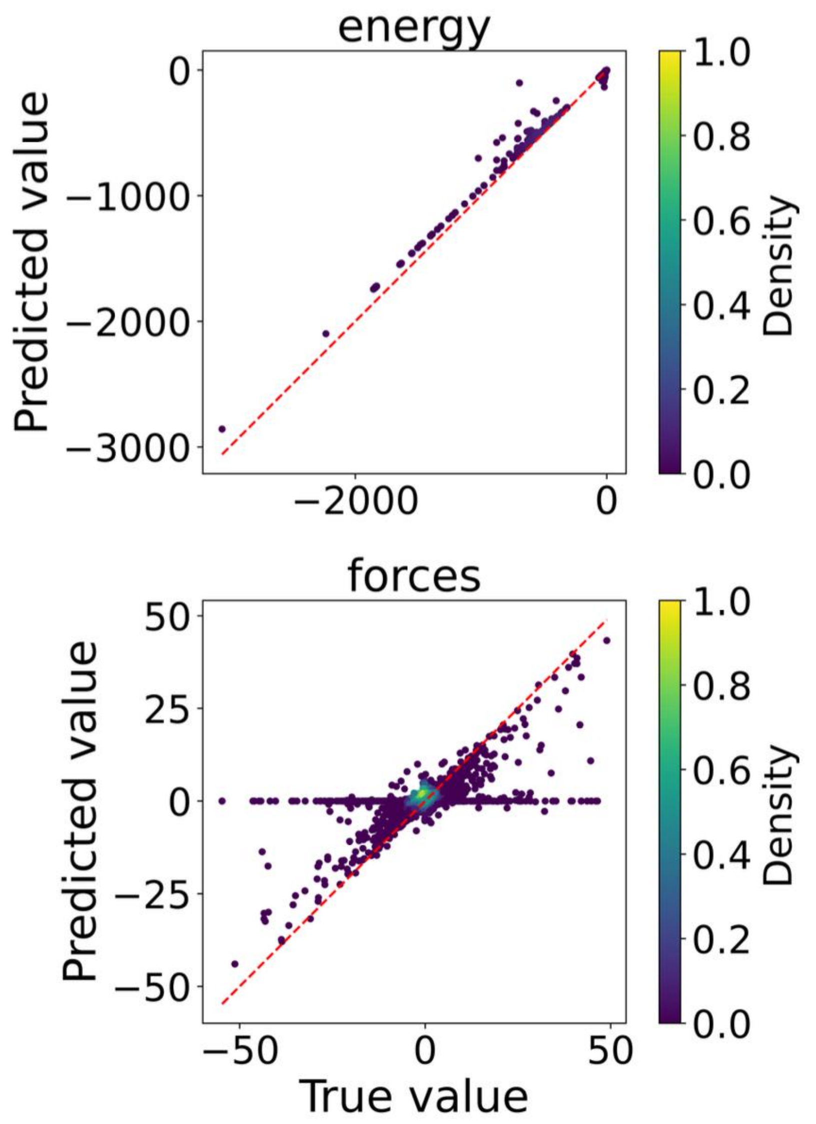
Ensemble Uncertainty Quantification



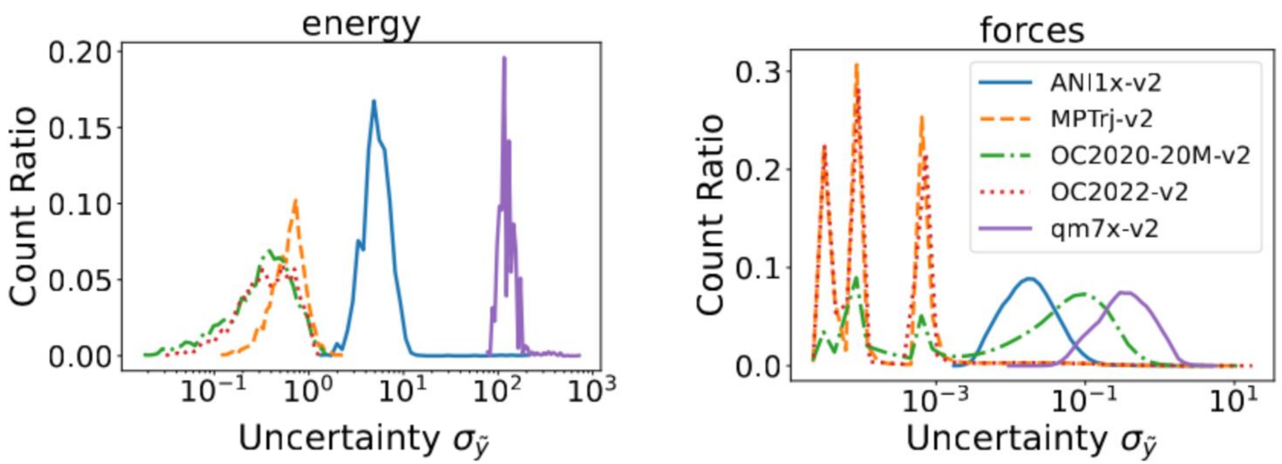


9. Scalable predictive graph foundation models (GFMs)

Dataset (Testing portion)	MAE energy (eV)	MAE forces (eV/angstrom)
QM7x	31.91	0.23
ANI1x	1.43	0.02
MPTrj	0.39	0.14
OC2020	0.10	0.11
OC2022	0.15	0.08



Uncertainty quantification



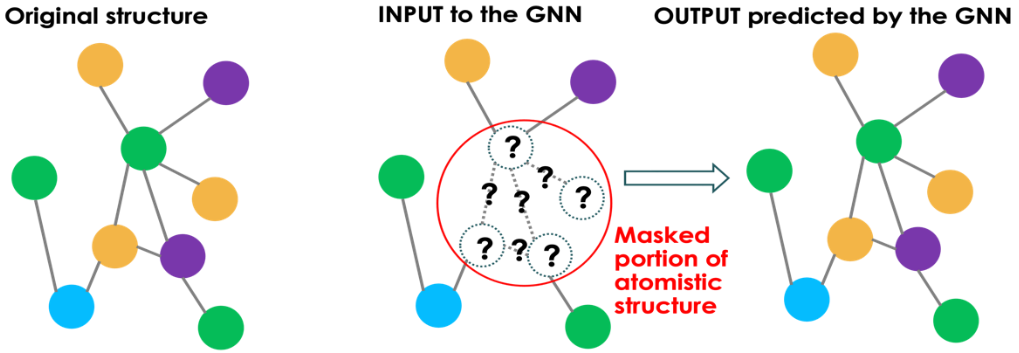
Oak Ridge National Laboratory

Future Work: Equivariant Generative Diffusion Models with Masking Techniques

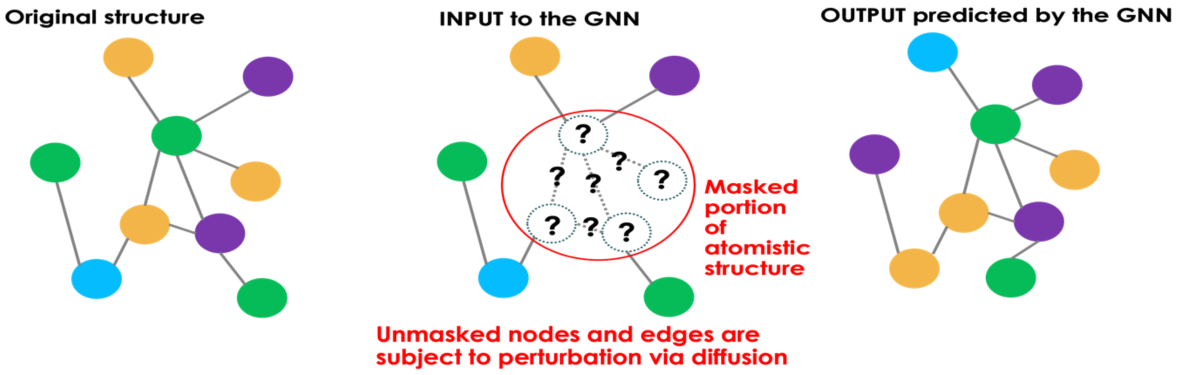
Approach: Perform autoregressive graph masking at each iterative step of the reversed diffusion process

- Impose **equivariance** constraints on the diffusion process to eliminate redundancies
- Use **global attention mechanisms** with GraphGPS to account for long-range interactions
- Use the **mask size as a tunable parameter**, to find the best compromise between exploration and exploitation

Standard Masking



Hybrid Diffusion + Masking



Expected outcome:

(a) **efficient exploration** of the material space

(b) the combined effect of equivariance + the robustness and computational efficiency of autoregressive graph masking will result in **computational and energy savings**

Future Work: Integrate Distributed Data Parallelism (DDP), Model Pipeline Parallelism (MPP), and Model Tensor Parallelism (TMP)

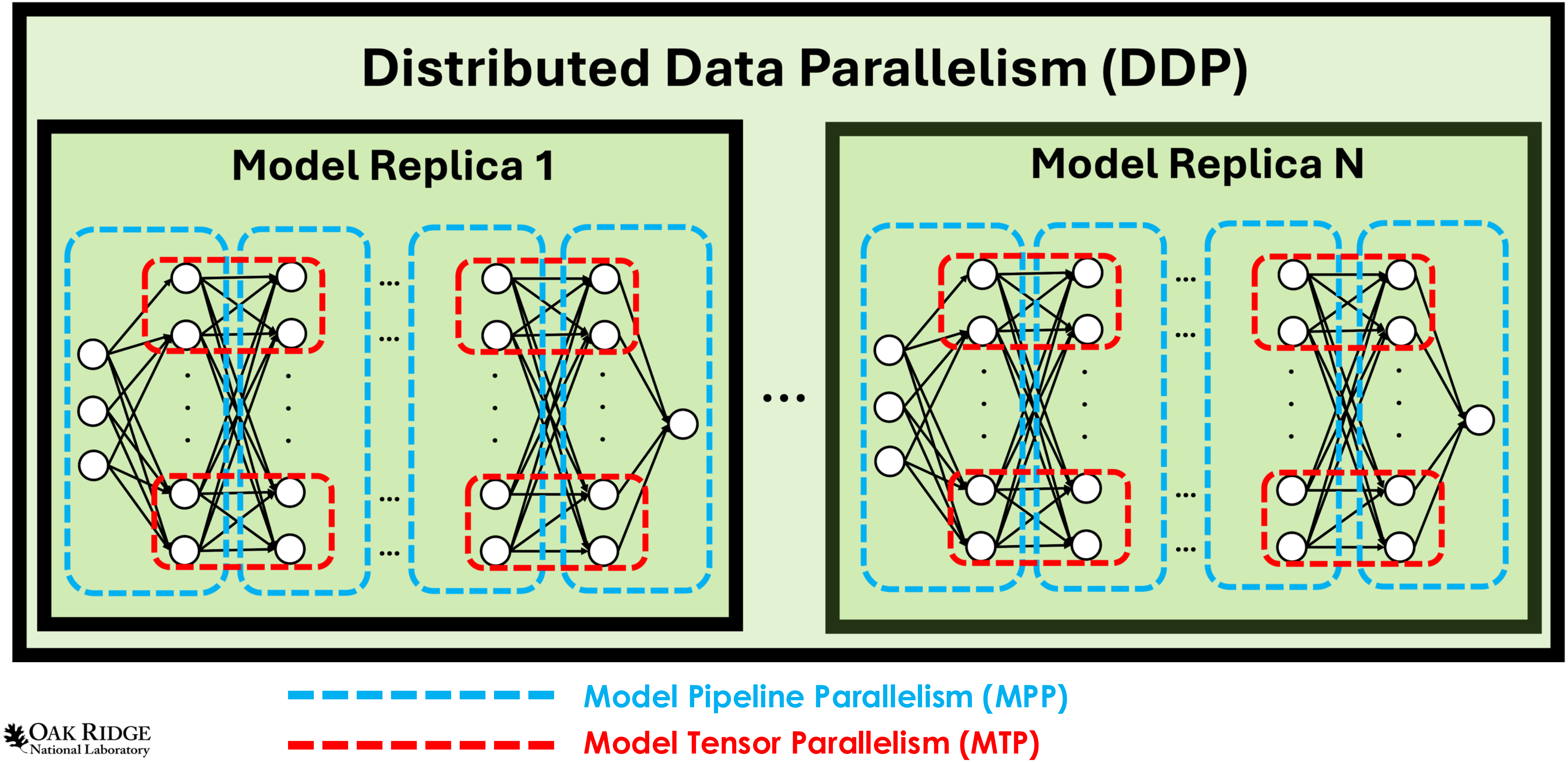
Approach: Hierarchically integrate DDP, MPP, and MTP

- If the data is too large, **DDP** will be used to partition the data across multiple GPUs.
- If the data has also broad variability, larger models may be needed to properly capture it. If the number of layers in the model is too large to fit in a single GPU, **MPP** will be used to split different layers across separate GPUs.
- If the number of neurons in each layer makes the model too large to fit in a single GPU, **MTP** will be used to split different neurons across separate GPUs.

Expected outcome:

- GNN architecture that simultaneously combines DDP, MPP, and MTP allowing training of **large, complex GNN models**
- Optimized latency and memory overhead and efficient training of GNN models at very large scale resulting in **energy savings**

Future Work: Integrate Distributed Data Parallelism, Model Pipeline Parallelism, and Model Tensor Parallelism (cont.'d)



Future Work: Generalize MTL for Stable and Energy Efficient Data Processing

<u>Approach</u>: Dedicate different heads of the GFM to process data from different sources.

- Ensure compatibility between implementation of hard parameter sharing and 3D parallelization
- New hard parameter sharing implementation:
  - uncovers the correlation between the data in the latent space,
  - sends to the heads only the data that each of them will process.

<u>Expected outcome</u>: Reduced number of parameters and unnecessary calculations, leading to:

(i) reduced computational resources and energy consumption; and

(ii) increased numerical stability and resilience against perturbations to the model's parameters