Optimizing Data Distribution and Kernel Performance for Efficient Training of Chemistry Foundation Models: A Case Study with MACE

Jesun Firoz

Future Computing Technologies Group

Pacific Northwest National Laboratory

HPDC 2025

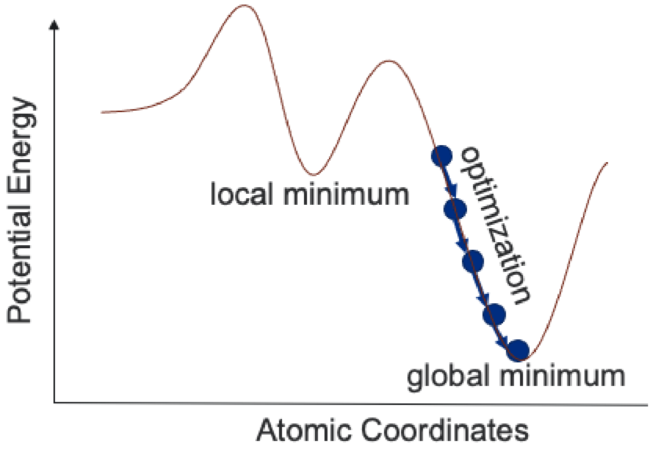
PNNL is operated by Battelle for the U.S. Department of Energy

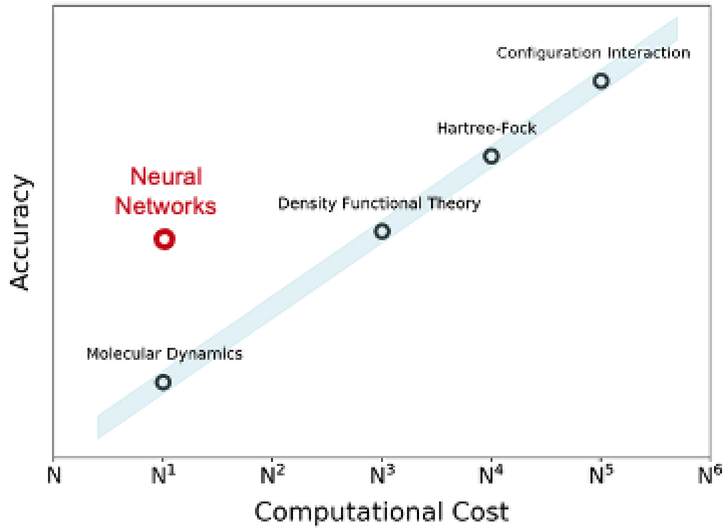
Authors

- Jesun Firoz (PNNL)
- Franco Pellegrini (SISSA)
- Mario Geiger (Nvidia)
- Darren Hsu (Nvidia)
- Jenna A. Bilbrey (PNNL)
- Han-Yi Chou (Nvidia)
- Maximilian Stadler (Nvidia)
- Markus Hoehnerbach (Nvidia)
- Tingyu Wang (Nvidia)
- Dejun Lin (Nvidia)
- Emine Kucukbenli (Nvidia)
- Henry W. Sprueill (PNNL)
- Ilyes Batatia (University of Cambridge)
- Sotiris S. Xantheas (PNNL)
- MalSoon Lee (PNNL)
- Chris Mundy (PNNL)
- Gabor Csanyi (University of Cambridge)
- Justin S. Smith (Nvidia)
- Ponnuswamy Sadayappan (University of Utah)
- Sutanay Choudhury (PNNL)

The Challenge in Simulation in Computational Chemistry

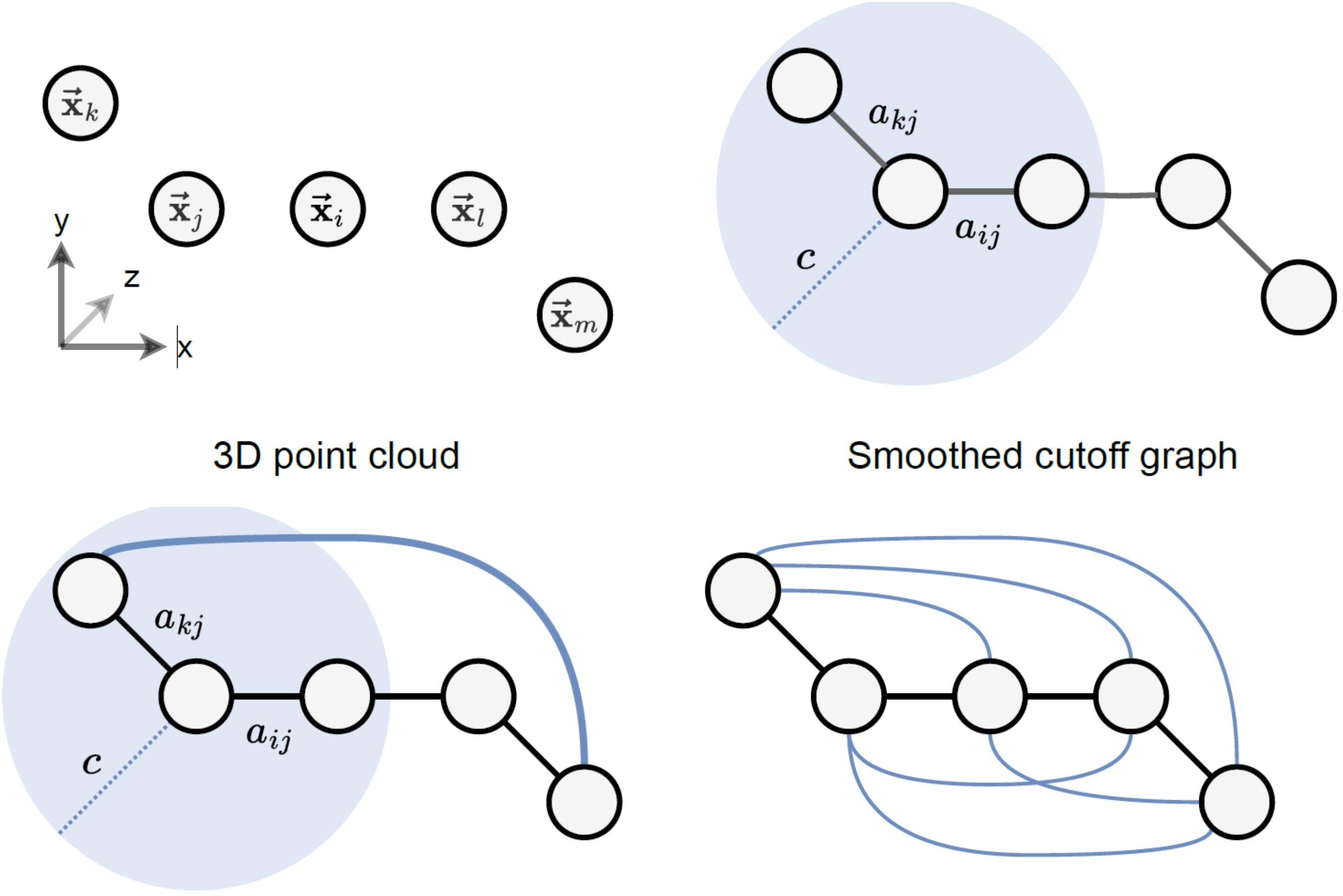
- Atomic simulations explore the potential energy surface (PES) of a molecule
- The PES relates **atomic positions** to a molecule’s **potential energy**
- Ab initio Molecular dynamics uses quantum methods (like **Density Functional Theory DFT**) to calculate the PES “on the fly” at each step, allowing accurate study of bond breaking/forming and reactive events.
- **Current Limitations of DFT:**
  - ➢ computationally expensive
  - ➢ Limited to nanosecond timescales
  - ➢ High time complexity
- **The Solution:**
  - ▪ **Machine Learning Interatomic Potentials (MLIPs)**
  - ▪ Faster alternatives to DFT calculations
  - ▪ Map atomic environments to energies and forces
  - ▪ Enable larger scale simulations





MLIP Input: Molecular Geometric Graphs

- Molecules can be represented as **geometric graphs**.



3D point cloud

Smoothed cutoff graph

Long-range connections

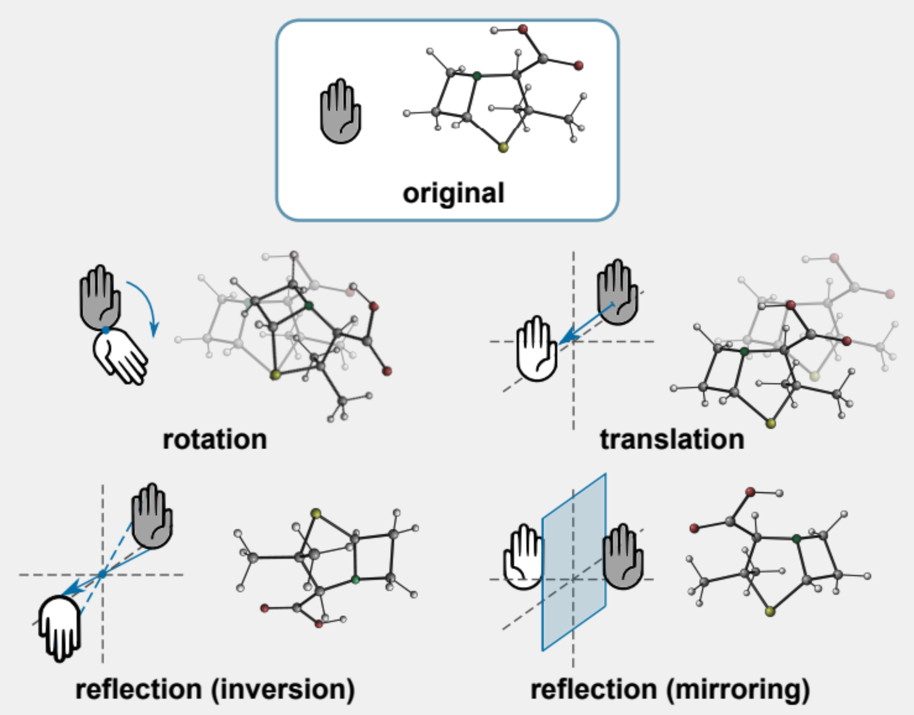
Complete graph

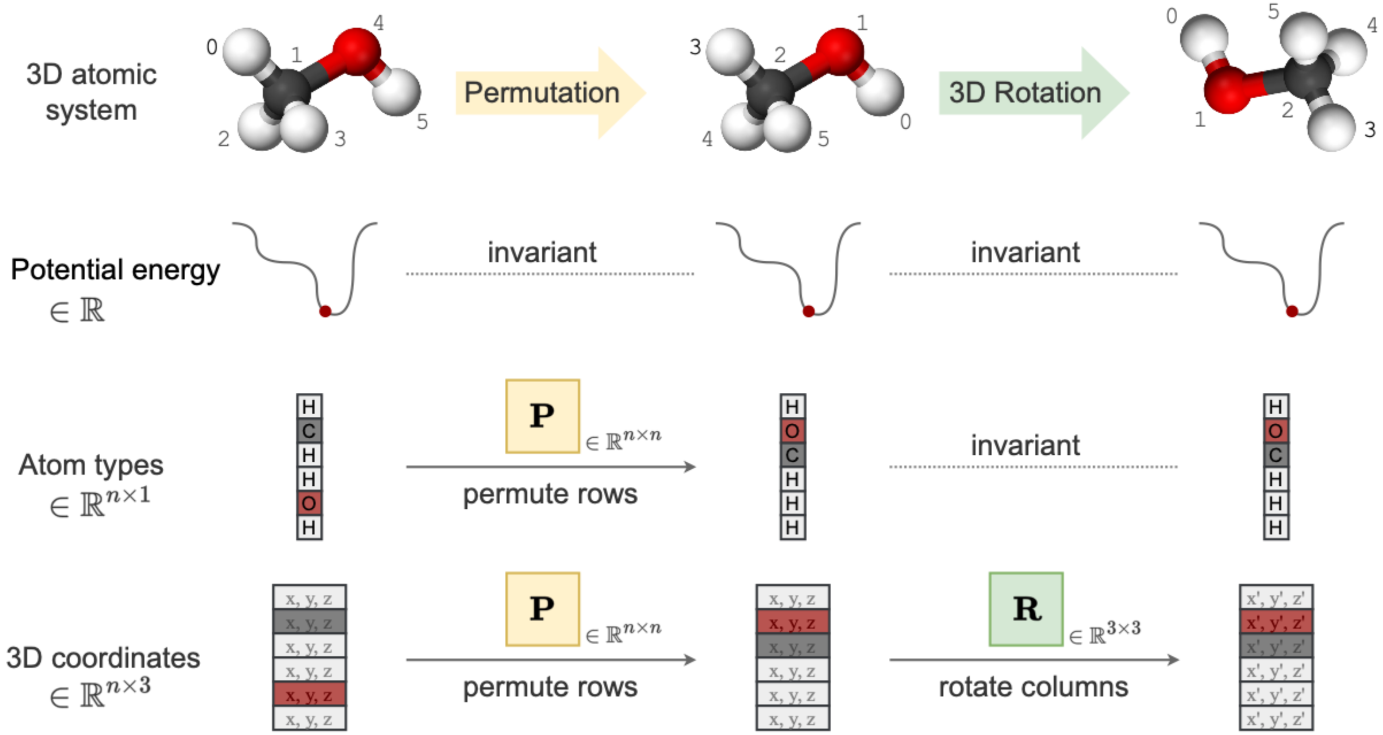
- In a geometric graph:
  - Nodes represent atoms embedded in 3D Euclidean space with **scalar attributes** (e.g. atom type) and **geometric attributes** (e.g. position, velocity, or forces).
- Edges are weighted by pairwise distances
- Smoothed cutoff graph most often applied
  - Cutoff *c* is a hyperparameter
  - Improves computational efficiency
- Long-range connections can be used to model periodic boundary conditions

"A Hitchhiker's Guide to Geometric GNNs for 3D Atomic Systems." (2023) arXiv:2312.07511

Molecular properties

- The **potential energy** of an isolated molecule remain the same no matter how the molecule is rotated or translated in space -- > **invariant** to Euclidean transformations.
- Rotating or translating the molecule will lead to an equivalent transformation of the directional **forces** acting on each atom; atomic forces are **equivariant** to Euclidean transformations.

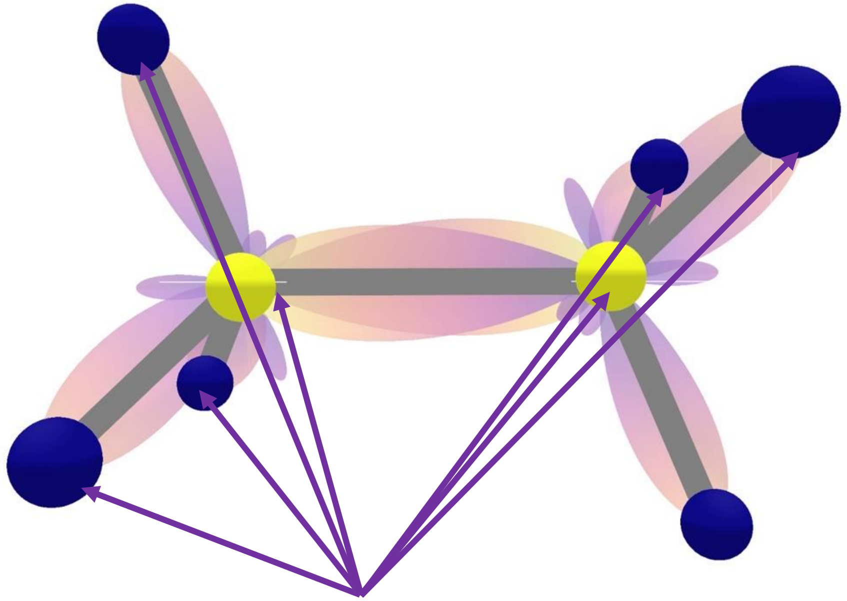



3D atomic system
0 1 2 3 4 5
Permutation
3 2 1 4 5 0
3D Rotation
0 5 4 1 2 3
Potential energy $\in \mathbb{R}$
invariant
invariant
Atom types $\in \mathbb{R}^{n\times 1}$
H C H H O H
P $\in \mathbb{R}^{n\times n}$
permute rows
H O C H H H
invariant
H O C H H H
3D coordinates $\in \mathbb{R}^{n\times 3}$
x, y, z
P $\in \mathbb{R}^{n\times n}$
permute rows
x, y, z
R $\in \mathbb{R}^{3\times 3}$
rotate columns
x', y', z'

"A Hitchhiker's Guide to Geometric GNNs for 3D Atomic Systems." (2023) arXiv:2312.07511

Representing the Features of an Atom in a Molecule

- In molecules, **spherical harmonics:**
  - describe how electrons are arranged and behave around atoms.
  - define the *angular* part of electron orbitals—regions where electrons are likely to be found.
- The shape of each atomic orbital is determined by a corresponding spherical harmonic function.
  - the “s” orbital is spherical (from the simplest spherical harmonic),
  - “p” orbitals are dumbbell-shaped, and
  - “d” orbitals have more complex, multi-lobed shapes.
- Specified by 2 parameters:
  - l (the "shape type" or "level of detail")
  - m (the "orientation" or "positioning")



Atoms [1]

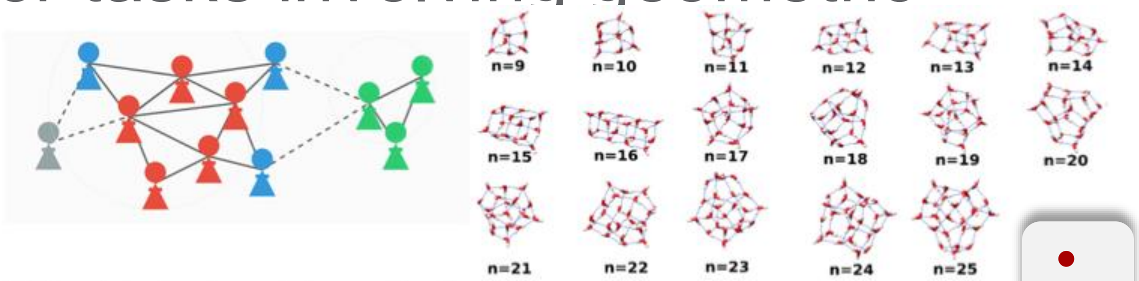
[1] Image courtesy: Tess Smidt https://docs.google.com/presentation/d/1Acz3YxUI-pH80n4UOeHWktnnpjsIsTXkiImP5ZGBNo4/edit?slide=id.g938c0ac37a_0_3097#slide=id.g938c0ac37a_0_3097

Important Terminologies

- **Spherical harmonics**:
  - express the angular features of atoms (directions, patterns, etc.).
- These features, organized as **spherical tensors**,
  - systematically encode information about directions and symmetries.
- When the system is rotated, the **Wigner D-matrix**
  - re-mixes the features in a way determined by the transformation law of spherical harmonics.
- When two features (or patterns) need to be **combined**—such as when considering interactions between two atoms—the **Clebsch–Gordan coefficients** specify exactly how their spherical harmonic expansions must be merged to form a new, valid angular feature.

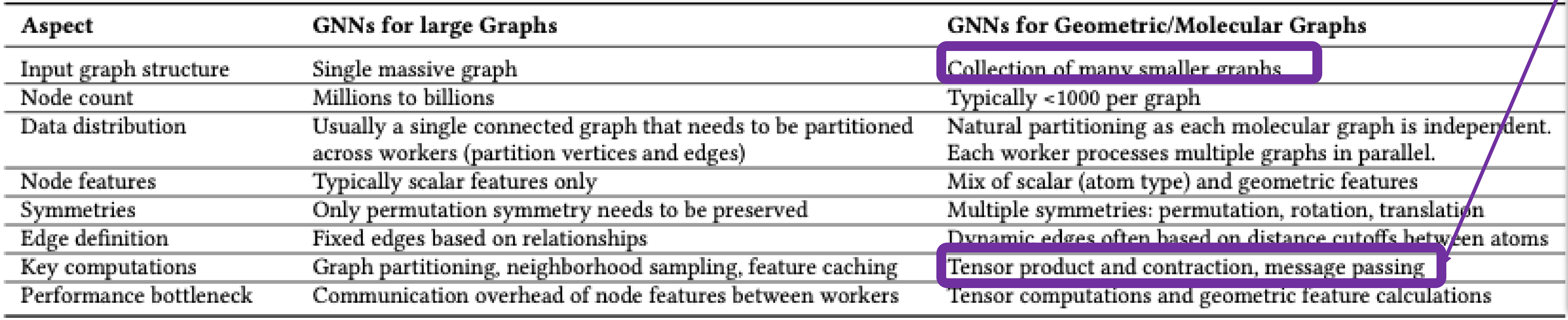
Comparing GNNs for Massive Graphs and Geometric/molecular Graphs

- Graph Neural Networks are suitable candidates as Surrogate model for MLIP.
- However, traditional GNNs are not well-suited for tasks involving geometric graphs.
- Need to adhere to the principles of physics.
- Important notion: A **symmetry** of an object
  - is a transformation of that object that leaves it unchanged
- Group theory allows us to formally describe and analyze these transformations. Equivariant GNNs apply these transformation internally to make sure that the model learns how to adjust “parameters” accordingly



(a) A social network graph (generally with many vertices and edges)

(b) An Ensemble of molecular graphs

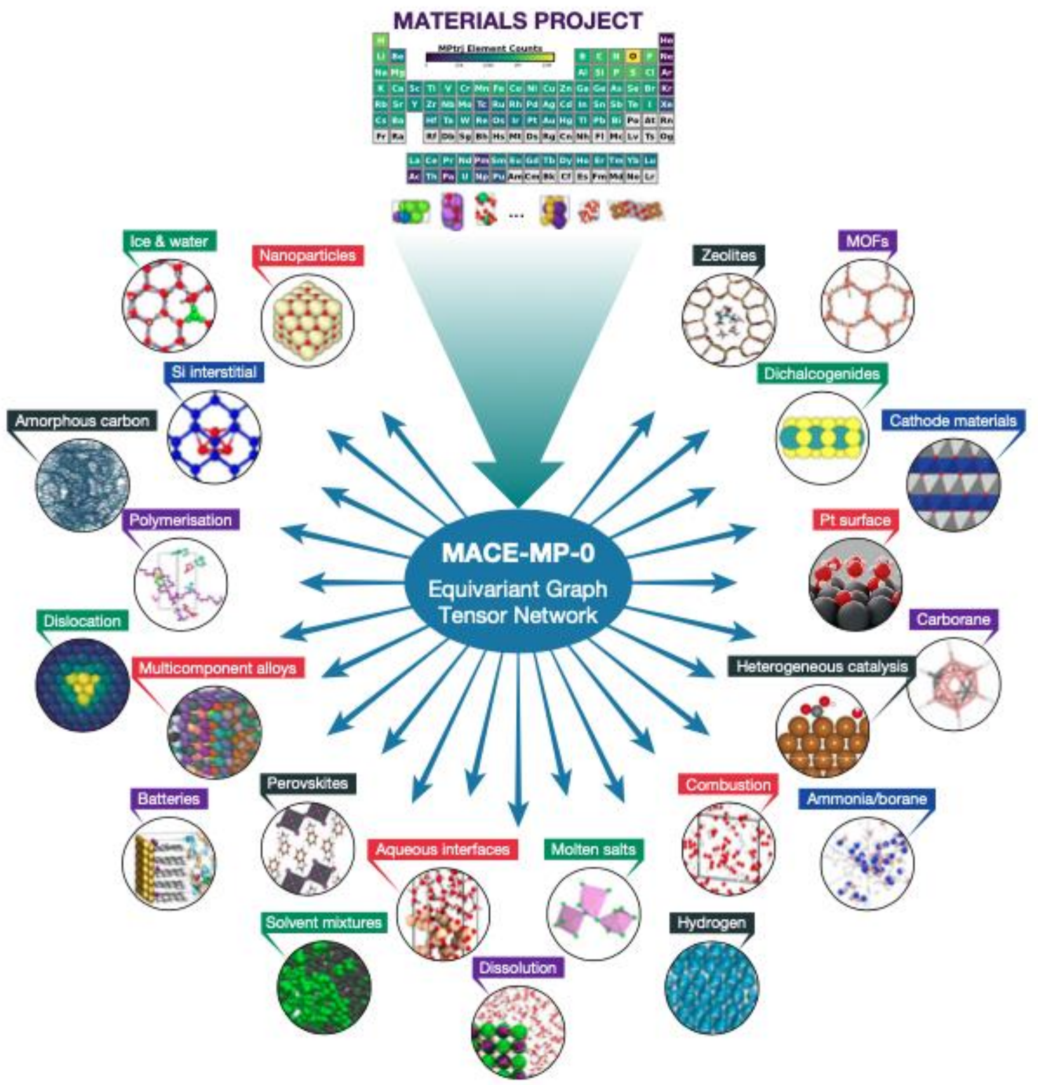


Aspect	GNNs for large Graphs	GNNs for Geometric/Molecular Graphs
Input graph structure	Single massive graph	Collection of many smaller graphs
Node count	Millions to billions	Typically <1000 per graph
Data distribution	Usually a single connected graph that needs to be partitioned across workers (partition vertices and edges)	Natural partitioning as each molecular graph is independent. Each worker processes multiple graphs in parallel.
Node features	Typically scalar features only	Mix of scalar (atom type) and geometric features
Symmetries	Only permutation symmetry needs to be preserved	Multiple symmetries: permutation, rotation, translation
Edge definition	Fixed edges based on relationships	Dynamic edges often based on distance cutoffs between atoms
Key computations	Graph partitioning, neighborhood sampling, feature caching	Tensor product and contraction, message passing
Performance bottleneck	Communication overhead of node features between workers	Tensor computations and geometric feature calculations

- Spherical tensors capture this

Chemistry Foundation Model

- **Traditional Approach:**
  - Specialized models for specific chemical systems for specific property prediction
  - Limited transferability
- **Foundation Models (Our Focus):**
  - Handle diverse chemical species
  - Capable of predicting diverse properties: force, energy, stress, ..
  - Enable zero-shot predictions
  - Support fine-tuning for specific applications
  - **Challenge**: Computational bottlenecks in training

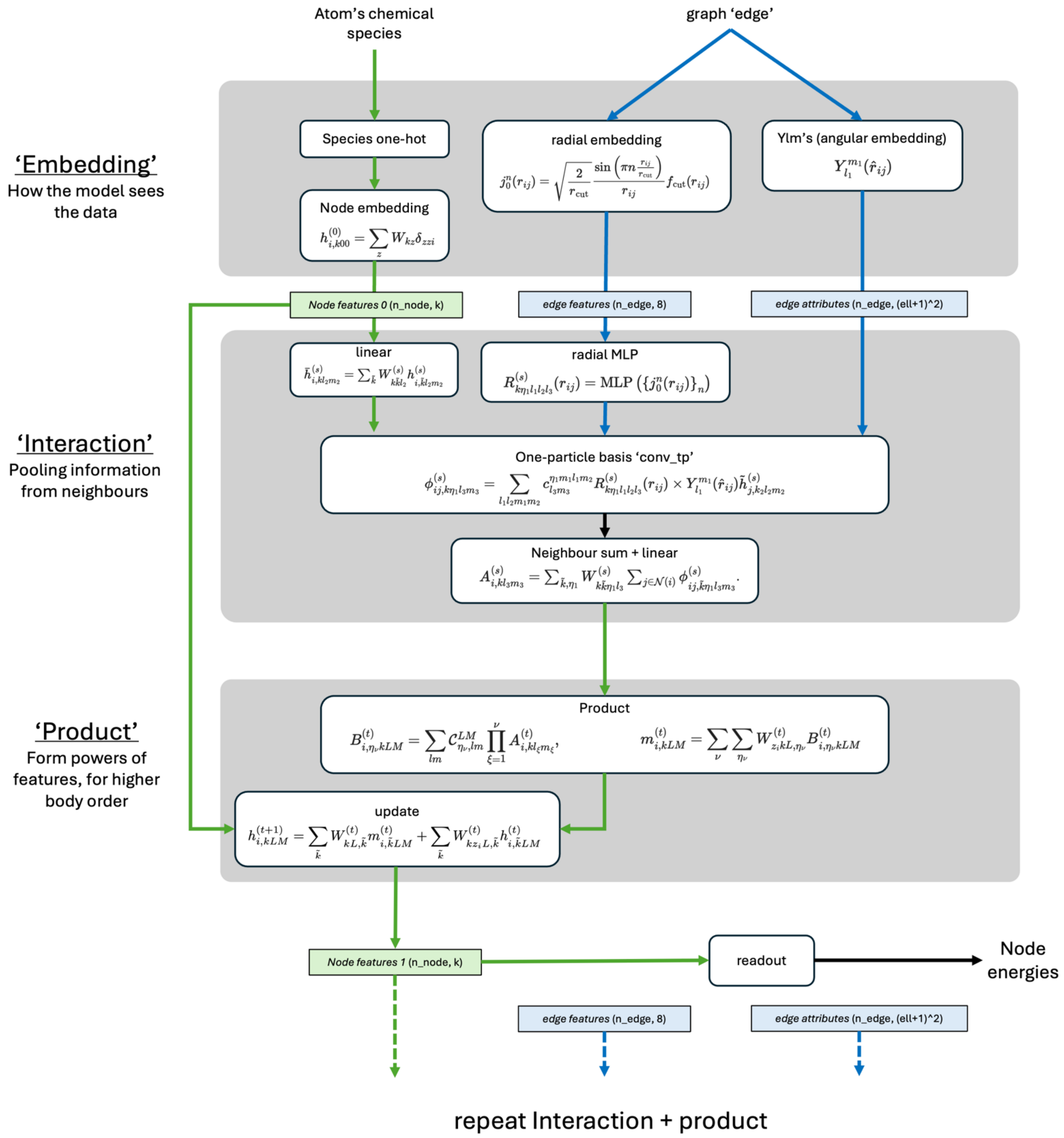


"A foundation model for atomistic materials chemistry." (2023) **arXiv:2401.00096**

Pacific Northwest
NATIONAL LABORATORY

MACE Architecture Overview

- **MACE: Higher Order Equivariant Message Passing Neural Networks**
  - State-of-the-art Chemistry Foundation Model (CFM)
  - Geometric Graph Neural Networks
- **Key Features:**
  - E(3)-equivariant (rotation and translation invariant)
  - Message passing between atoms
  - Higher-order tensor operations
- **Training Challenge:** Efficient scaling to large datasets and multiple GPUs



Atom's chemical species

graph 'edge'

'Embedding'
How the model sees the data

Species one-hot

Node embedding
$h^{(0)}_{i,k00} = \sum_z W_{kz} \delta_{zz_i}$

radial embedding
$j^n_0(r_{ij}) = \sqrt{\frac{2}{r_{\text{cut}}}} \frac{\sin\left(\pi n \frac{r_{ij}}{r_{\text{cut}}}\right)}{r_{ij}} f_{\text{cut}}(r_{ij})$

Ylm's (angular embedding)
$Y^{m_1}_{l_1}(\hat{r}_{ij})$

Node features 0 (n_node, k)

edge features (n_edge, 8)

edge attributes (n_edge, (ell+1)^2)

'Interaction'
Pooling information from neighbours

linear
$\bar{h}^{(s)}_{i,kl_2m_2} = \sum_{\tilde{k}} W^{(s)}_{k\tilde{k}l_2} h^{(s)}_{i,\tilde{k}l_2m_2}$

radial MLP
$R^{(s)}_{k\eta_1 l_1 l_2 l_3}(r_{ij}) = \text{MLP}\left(\{j^n_0(r_{ij})\}_n\right)$

One-particle basis 'conv_tp'
$\phi^{(s)}_{ij,k\eta_1 l_3 m_3} = \sum_{l_1 l_2 m_1 m_2} c^{\eta_1 m_1 l_1 m_2}_{l_3 m_3} R^{(s)}_{k\eta_1 l_1 l_2 l_3}(r_{ij}) \times Y^{m_1}_{l_1}(\hat{r}_{ij}) \tilde{h}^{(s)}_{j,k_2 l_2 m_2}$

Neighbour sum + linear
$A^{(s)}_{i,kl_3m_3} = \sum_{\tilde{k},\eta_1} W^{(s)}_{k\tilde{k}\eta_1 l_3} \sum_{j \in \mathcal{N}(i)} \phi^{(s)}_{ij,\tilde{k}\eta_1 l_3 m_3}.$

'Product'
Form powers of features, for higher body order

Product
$B^{(t)}_{i,\eta_\nu kLM} = \sum_{lm} \mathcal{C}^{LM}_{\eta_\nu,lm} \prod_{\xi=1}^{\nu} A^{(t)}_{i,kl_\xi m_\xi},$
$m^{(t)}_{i,kLM} = \sum_{\nu} \sum_{\eta_\nu} W^{(t)}_{z_i kL,\eta_\nu} B^{(t)}_{i,\eta_\nu kLM}$

update
$h^{(t+1)}_{i,kLM} = \sum_{\tilde{k}} W^{(t)}_{kL,\tilde{k}} m^{(t)}_{i,\tilde{k}LM} + \sum_{\tilde{k}} W^{(t)}_{kz_iL,\tilde{k}} h^{(t)}_{i,\tilde{k}LM}$

Node features 1 (n_node, k)

readout

Node energies

edge features (n_edge, 8)

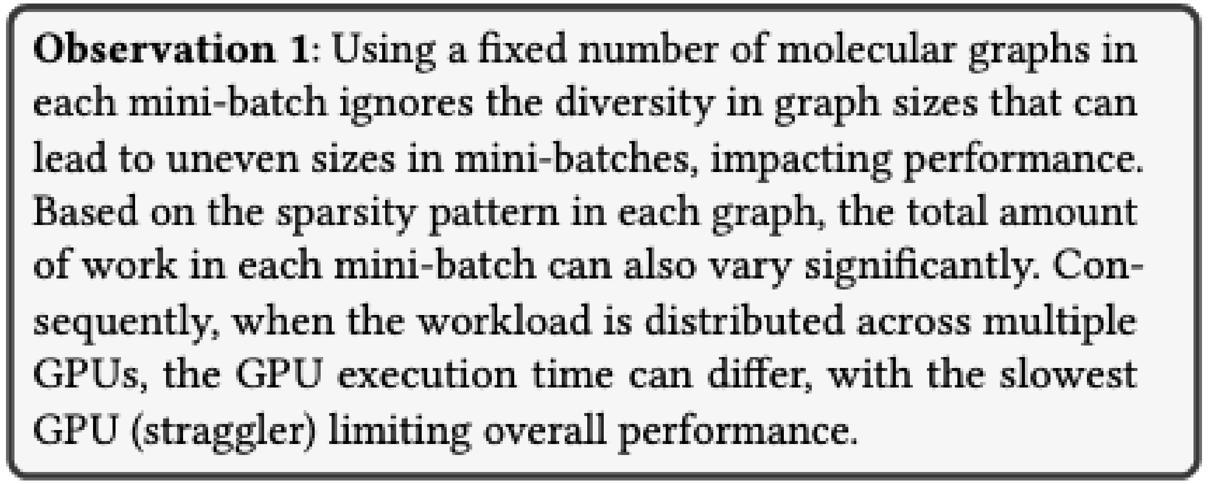
edge attributes (n_edge, (ell+1)^2)

repeat Interaction + product

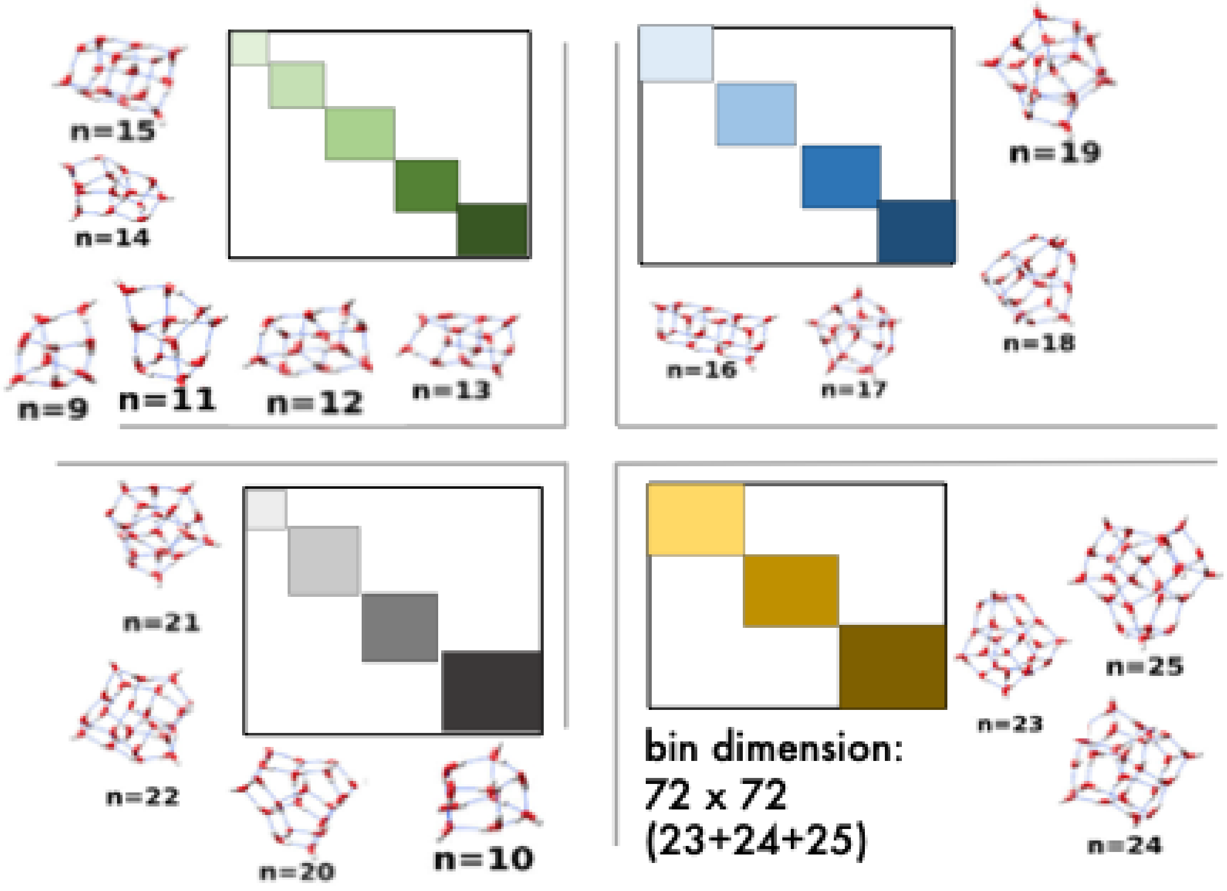
MACE tutorial: https://colab.research.google.com/drive/1AlfjQETV_jZ0JQnV5M3FGwAM2SGCl2aU#scrollTo=uEf7_Of6_03A

Challenges in Molecular GNN (1): Data Distribution

- **Load Imbalance due to Data Distribution**



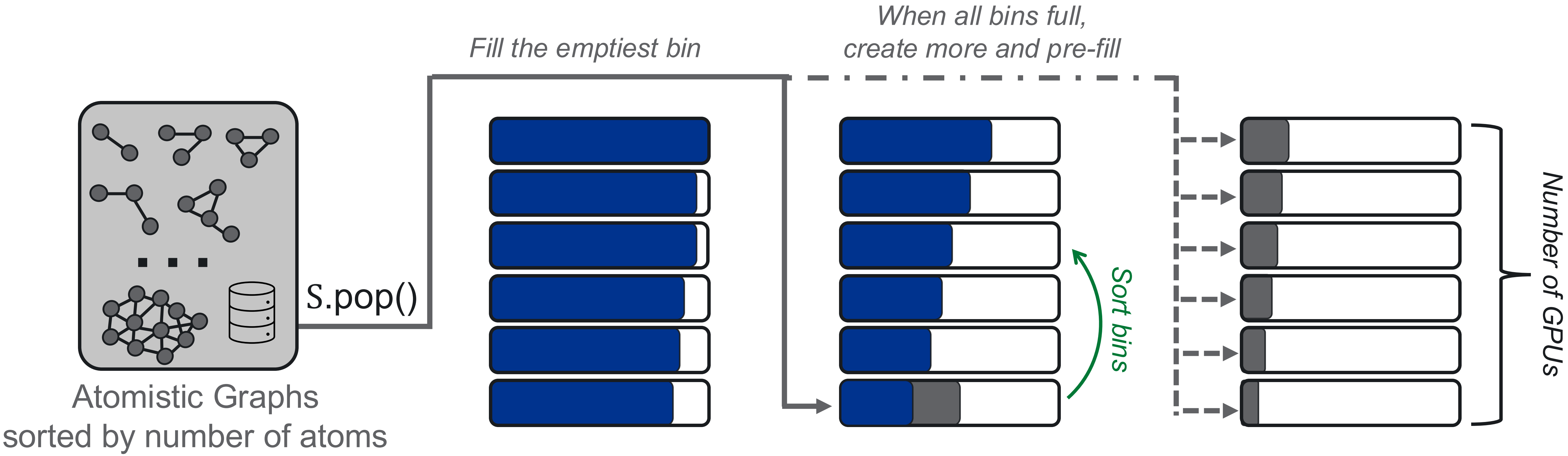
Observation 1: Using a fixed number of molecular graphs in each mini-batch ignores the diversity in graph sizes that can lead to uneven sizes in mini-batches, impacting performance. Based on the sparsity pattern in each graph, the total amount of work in each mini-batch can also vary significantly. Consequently, when the workload is distributed across multiple GPUs, the GPU execution time can differ, with the slowest GPU (straggler) limiting overall performance.



Data Distribution and Multi Objective Bin packing

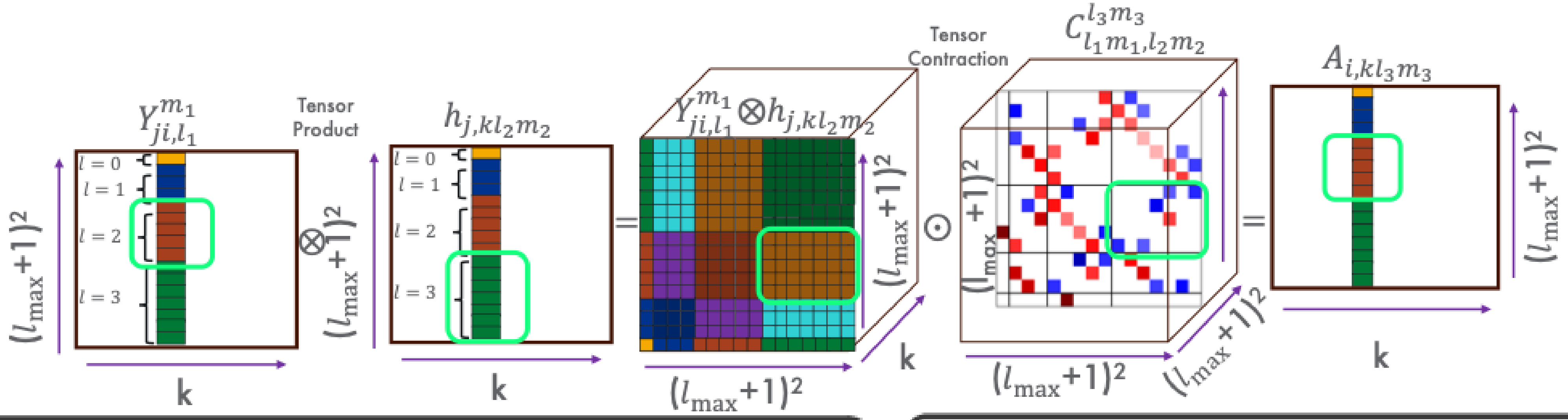
- **The Load Balancing Problem:**
  - Chemical systems vary significantly in size (9-768 atoms)
  - Uneven computational loads across GPUs
  - Need balanced distribution for optimal performance
- **Our Approach:**
  - Formulate as multi-objective bin packing problem with the following objectives:
    - ✓ minimize the number of bins,
    - ✓ while minimizing zero-padding memory in each bin (mini-batch), and
    - ✓ the difference between the amount of space filled by molecular graphs in any two bins should be minimal.
  - Iterative algorithm for practical solution
  - Fast and effective load balancing

Our Algorithm for Data Distribution



Challenges in Molecular GNN (2): Bottleneck in Computation

- **Kernel Performance Optimization**
  - **Problem**: Computational bottlenecks in training
  - Symmetric tensor contraction: core operation in MACE, NeuqIP, Allegro
    - ✓ High computational intensity
    - ✓ Critical for overall performance



Observation 2: The existing approach doesn't consider the sparsity of Clebsch-Gordan coefficients, leading to unnecessary computations in dense matrix multiplication, while exploiting these properties could significantly reduce storage and computational requirements.

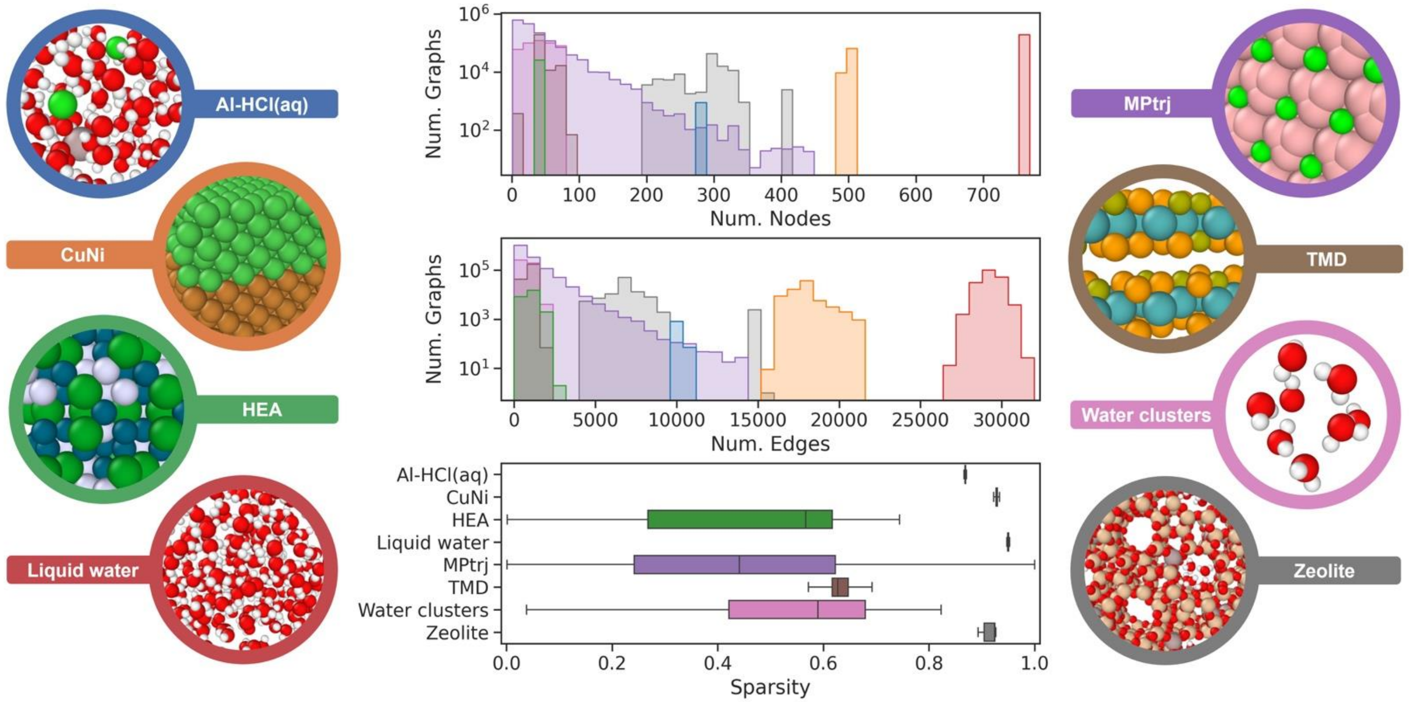
Observation 3: Existing implementations in frameworks like PyTorch e3nn perform symmetric tensor contractions by breaking them into many small separate kernel calls for each combination of quantum numbers (l,m), leading to excessive global memory access, poor GPU utilization, and frequent small kernel launch overhead.

Optimization for Symmetric Tensor Contraction

- **Key Computational Kernel:**
  - **Symmetric Tensor Contraction** - core operation in MACE, NequIP, Allegro
  - High computational intensity
  - Critical for overall performance
- **Optimization Strategies:**
  - Kernel fusion:
    - ✓ keeping intermediate results in shared memory as long as possible
  - Set of key rules that determine which combinations give non-zero CG coefficients.
    - ✓ pre-compute all valid combinations, store only non-zero coefficients, and create lookup tables for fast access
    - ✓ Sparse multiplications focusing only on non-zero elements
  - Vectorized loads through float4 operation
  - CUDA-based optimizations
    - ✓ Warp-level operations
    - ✓ Butterfly exchange patterns using __shfl_xor_sync()
    - ✓ Loop unrolling for better instruction scheduling

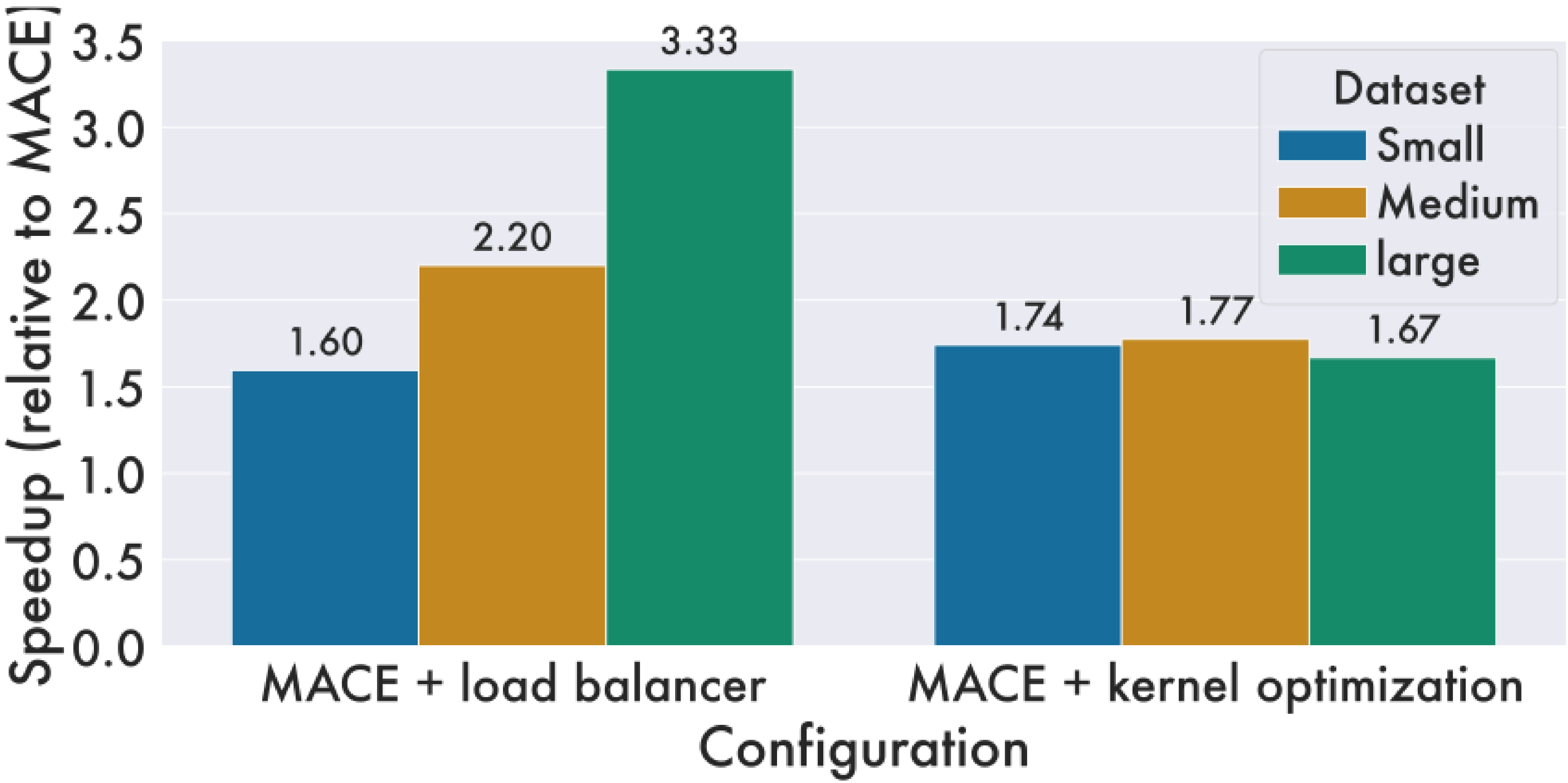
Experimental Setup

- **Dataset Characteristics:**
  - **Size**: 2.6M samples
  - **Diversity**: 8 different chemical systems
    - ✓ Water clusters (9-75 atoms)
    - ✓ Liquid water (768 atoms)
    - ✓ Metals (CuNi)
    - ✓ High-entropy alloys (HEA)
    - ✓ Transition metal dichalcogenides (TMD)
    - ✓ Zeolites

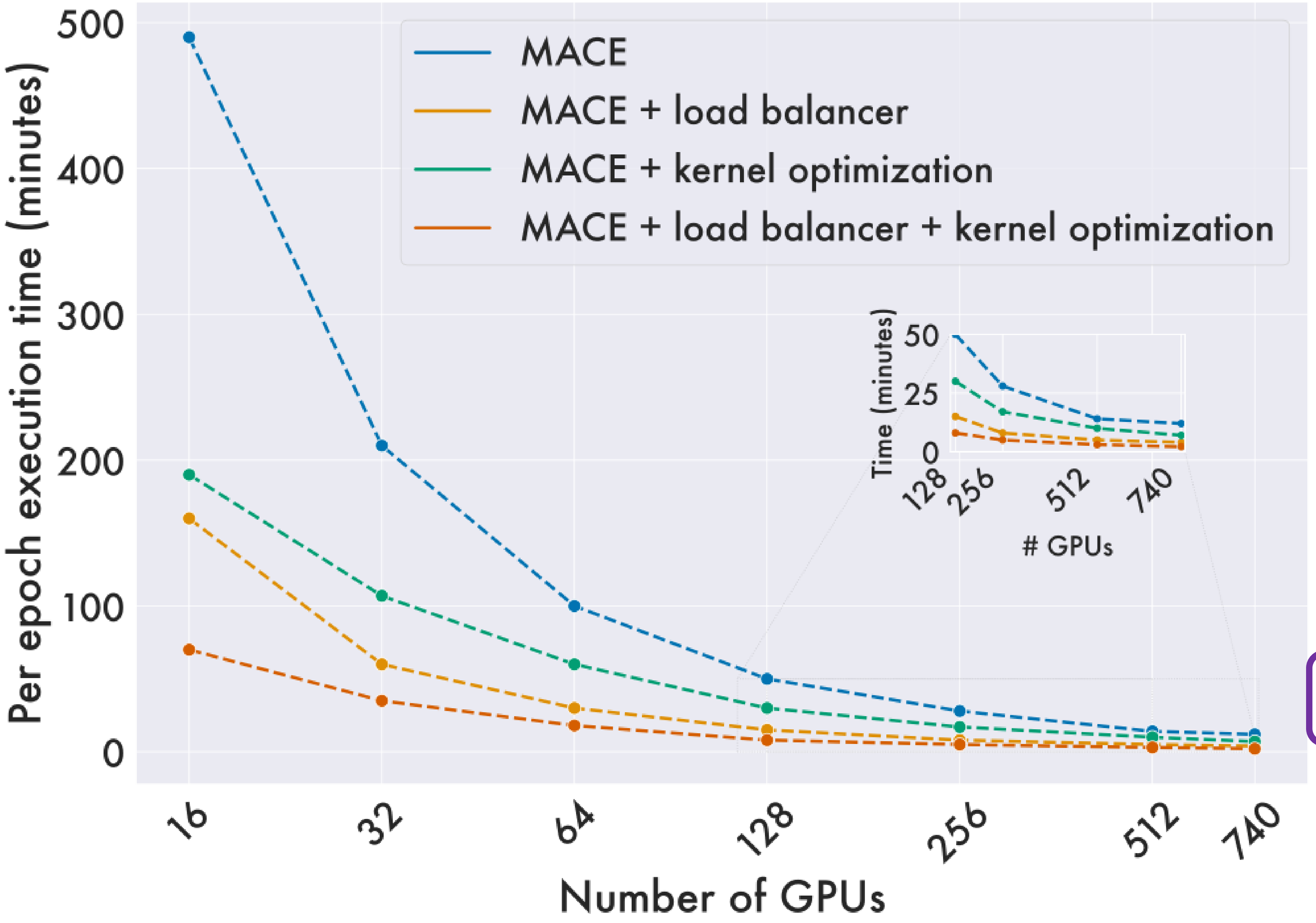


- **Hardware**: 740 40GB Nvidia A100 GPUs @ NERSC

Experimental Results: Ablation Study

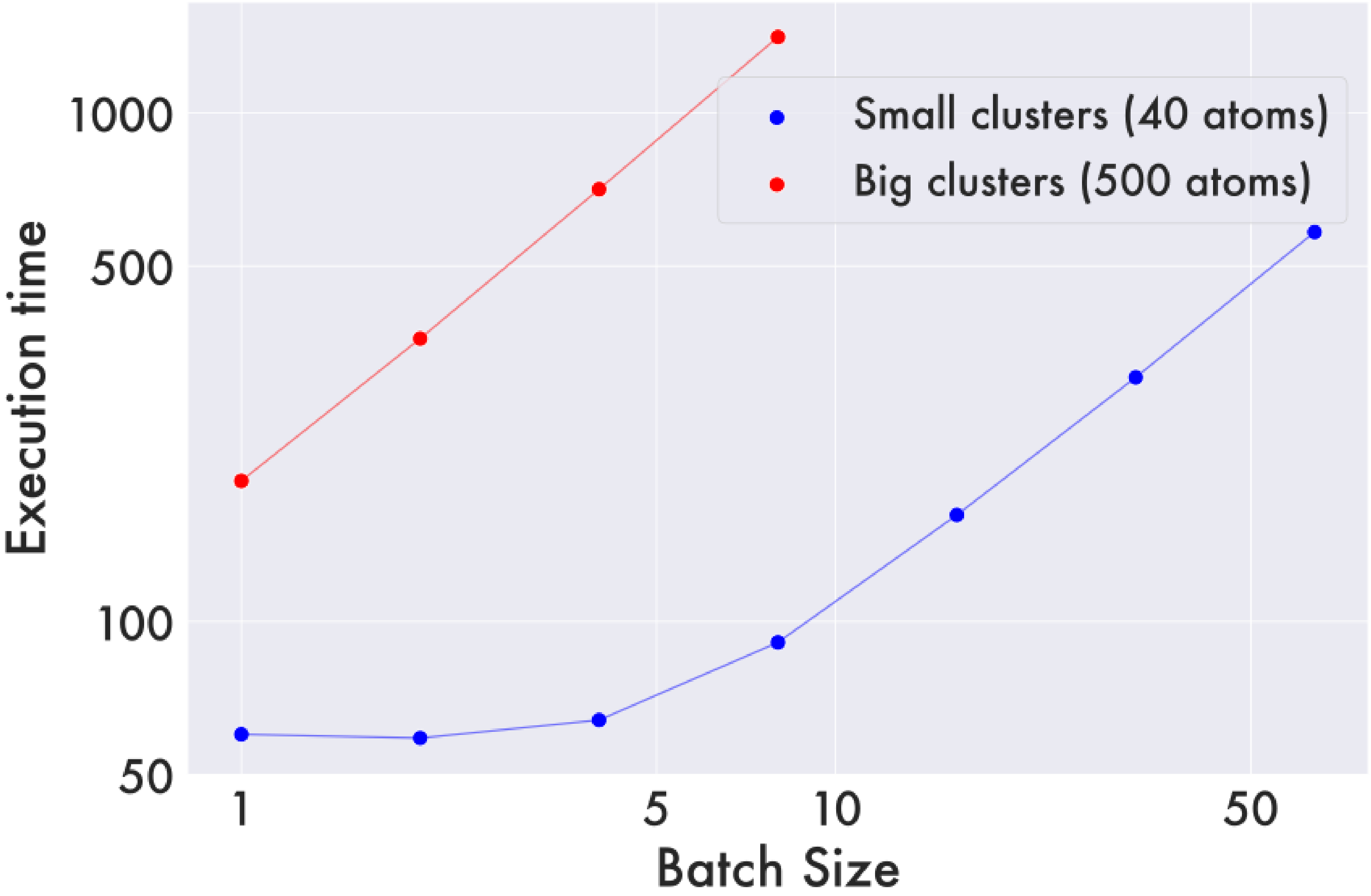


Experimental Results: Strong Scaling



6x speedup

Empirical Determination of Optimal Bin Capacity and Mini-batch Size



Conclusion

- Addressed critical scaling bottlenecks in chemistry foundation model training
    - Molecular datasets exhibit extreme heterogeneity in terms of system sizes, atom types, and sparsity patterns
    - Requires careful data distribution to avoid load imbalance and straggler effects across GPUs.
    - Symmetric tensor contractions—high-cost operations central to equivariant GNNs
- Developed practical solutions for load balancing and kernel optimization
- Demonstrated substantial performance improvements (6x speedup).

Thank you