BRANCHIP: LEARNING ADAPTIVE EQUIVARIANT COMPUTATION FOR INTERATOMIC POTENTIALS

Anonymous authors
Paper under double-blind review

ABSTRACT

Equivariant machine learning interatomic potentials (MLIPs) have revolutionized atomistic modeling, but accurate treatment of complex materials and molecular systems demands expensive models. This limits simulation length- and time-scales, with tensor products a key computational bottleneck. The recent emergence of foundation-scale MLIPs further exacerbates this challenge. We present Branch Interatomic Potential (BranchIP), a single-model framework for learned adaptive tensor product computation, trained with a novel distillation loss. In our experiments on two systems of physical interest, a heterogeneous catalysis system and a proton-conducting solid acid electrolyte, BranchIP accelerates MLIPs across model sizes by up to $2.4\times$ while reducing memory usage by up to $2.6\times$. This is achieved while maintaining physical fidelity. Furthermore, the learned adaptive computation provides model interpretability by revealing which interactions demand deeper computation and showing how computational depth relates to chemical complexity and dynamics.

1 INTRODUCTION

Machine learning interatomic potentials (MLIPs) enable near first-principles modeling of diverse materials and molecules ranging from heterogeneous catalysts to biomolecular systems (Behler & Parrinello, 2007; Batzner et al., 2022; Batatia et al., 2022; Owen et al., 2024; Kozinsky et al., 2023). Practical applications, such as molecular dynamics (MD) simulations, require repeated evaluations of MLIPs, demanding both high predictive accuracy and fast inference (Frenkel & Smit, 1996). Like vision and language models, MLIPs benefit from deep neural network (NN) architectures that progressively refine feature representations through successive nonlinear layers. The resulting deep MLIPs are largely considered black boxes and their interpretability remains an open challenge (Kalita et al., 2025; Zhang et al., 2025; Creed et al., 2026).

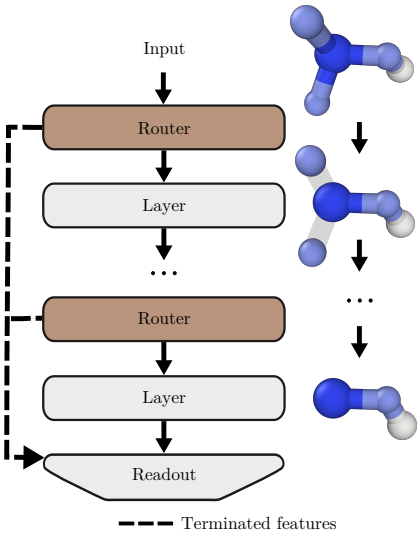


Figure 1: Overview of BranchIP. Routers placed before model layers terminate sufficiently refined features to reduce computation. Features represent local atomic interactions.

The major breakthroughs in language and vision models from scaling dataset and model sizes (Kaplan et al., 2020; Hoffmann et al., 2022; Zhai et al., 2022) recently motivated analogous efforts for MLIPs, resulting in pretrained foundation models (FMs) that generalize across diverse chemistry domains (Chen & Ong, 2022; Park et al., 2024; Wood et al., 2026; Batatia et al., 2025; Kavanagh et al., 2026). However, computational cost hinders the practical use of large-scale MLIPs. The dominant bottleneck is the tensor product (TP), an operation which combines features from neighboring atoms to build the many-body geometric information essential for high accuracy. Most acceleration efforts reduce the cost of individual TPs through efficient formulations or optimized implementations (Passaro & Zitnick, 2023; Luo et al., 2024; Lee et al., 2025; Bharadwaj et al., 2025). Recently, dynamic cutoffs have reduced the number of TPs by adapting the interaction graph (Han et al., 2026). This

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

aligns with broader efforts in adaptive inference that combine potentials of different cost or fidelity (Birks et al., 2026; Wood et al., 2026; Immel et al., 2025). These methods exploit the heterogeneous chemical complexity of realistic atomistic systems; only a subset of atomic environments requires deeper and more expensive architectures for accurate modeling. Existing approaches typically train and evaluate several independent models, resulting in expensive and complex training schemes with exorbitant parameter counts. Furthermore, assigning atomic environments to different models relies on hand-crafted descriptors that require prior knowledge of system dynamics—the very dynamics atomistic modeling often seeks to discover.

To address these limitations, we introduce Branch Interatomic Potentials (BranchIPs), a single-model, learned adaptive computation framework for equivariant MLIPs inspired by early exits in deep NNs (Teerapittayanon et al., 2016) (Figure 1). As in classification, where early exits were first introduced, some inputs can be handled by shallower networks, while others require the full depth. Rather than applying the TP to every feature at every layer, BranchIP removes features from subsequent TPs once further refinement no longer improves accuracy. The routing patterns reveal the TP refinement, usually representative of the body-order of model representations, required for various atomic environments. This improves the interpretability of the hidden model layers.

We implement BranchIP in the state-of-the-art equivariant MLIP Allegro (Musaelian et al., 2023; Tan et al., 2026), termed Branchlegro. We demonstrate Branchlegro on two systems: a heterogeneous catalysis system comprising platinum surfaces with gaseous and adsorbed carbon monoxide (CO/Pt) (Owen et al., 2024) and proton transport in the superprotonic solid acid electrolyte CsH_2PO_4 (CDP), a system without an obvious chemically motivated spatial partition (Wang et al., 2025). For both systems, deriving chemical insight requires MD simulations over large length- and time-scales to sufficiently sample rare events, making computational efficiency essential.

Our main contributions are:

1. We develop a single-model framework for **feature-level adaptive TP** computation in MLIPs.
2. We introduce a **fully learned dynamic router**, trained with a novel distillation objective, that smoothly routes features through TP computation obeying exact energy conservation.
3. We implement the method in Allegro and prove it retains a twice continuously differentiable potential energy surface (PES) as well as equivariance.
4. We demonstrate the method on bespoke and foundation MLIPs for CO/Pt and CDP systems, **accelerating inference and reducing memory usage** with limited degradation in accuracy.
5. The learned routing provides model interpretability, revealing **which interactions require deeper equivariant computation** and how this varies with chemistry and dynamics.

2 Background and related work

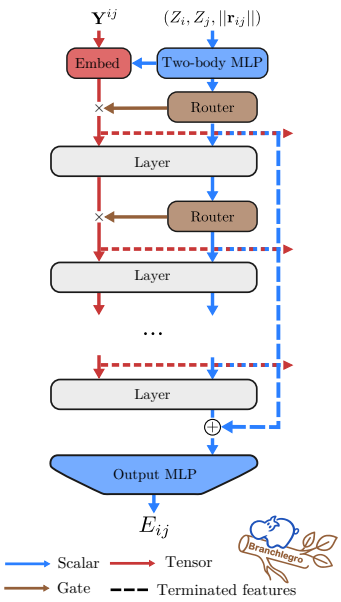
MLIPs use NNs trained on reference data, usually from expensive quantum methods, to approximate the PES. Given atomic positions and chemical species, $\{\mathbf{r}_i, Z_i\}$, as input, the model predicts a total energy, E. Automatic differentiation then yields forces and potentially stresses.

Allegro (Musaelian et al., 2023) is a strictly-local, equivariant deep NN architecture for interatomic potentials. The model uses only operations that are equivariant under the three-dimensional Euclidean group, $E(3)$, consisting of rotations, translations, and reflections. As a result, the network predictions transform consistently with the underlying atomic geometry. Allegro decomposes the total energy into pair contributions from neighboring atoms, where neighbors are defined by a fixed cutoff distance. Tensor and scalar features represent each ordered atom pair, and an output multilayer perceptron (MLP) predicts the pair energy from the scalar features. At each layer, a TP combines features from neighboring pairs to incorporate many-body geometric information while preserving equivariance.

TP accelerations largely focus on cheaper formulations (Xu et al., 2026; Passaro & Zitnick, 2023; Luo et al., 2024) and optimized kernels (Lee et al., 2025; Tan et al., 2026; Bharadwaj et al., 2025) and are complementary to BranchIP.

Adaptive interatomic potential architectures seek to reduce inference cost while maintaining accuracy. **ML-MIX** relies on a user-defined assignment of expensive and inexpensive interatomic potentials to predefined spatial regions (Birks et al., 2026). Blending forces from different models

(a) Branchlegro schematic.



(b) Pseudocode of the Branchlegro inference.

BRANCHLEGRO FORWARD PASS

Require: Edge set $\mathcal{E}$; embedding $(\mathbf{x}^{ij,0}, w^{ij,0}, \bar{\mathbf{V}}^{ij,0}, \mathbf{Y}^{ij})$.

$\mathcal{E}_0 \leftarrow \mathcal{E}$
for $l = 1, \ldots, L$ **do**
  if $l-1$ has a router **then**
    $f^{ij,l-1} \leftarrow \mathcal{F}\big(\mathrm{MLP}_{\mathrm{router}_{l-1}}(\mathbf{x}^{ij,l-1})\big)\,,\ ij \in \mathcal{E}_{l-1}$
    $\mathbf{V}^{ij,l-1} \leftarrow f^{ij,l-1}\bar{\mathbf{V}}^{ij,l-1},\ ij \in \mathcal{E}_{l-1}$
    $\mathcal{E}_l \leftarrow \{ij \in \mathcal{E}_{l-1} : f^{ij,l-1} > 0\}$
  else
    $\mathbf{V}^{ij,l-1} \leftarrow \bar{\mathbf{V}}^{ij,l-1},\ ij \in \mathcal{E}_{l-1}$
    $\mathcal{E}_l \leftarrow \mathcal{E}_{l-1}$
  $\bar{\mathbf{V}}^{ij,l} \leftarrow \left(\sum_{k:(i,k)\in\mathcal{E}} w^{ik,l-1}\mathbf{Y}^{ik}\right) \otimes \mathbf{V}^{ij,l-1},\ ij \in \mathcal{E}_l$
  $\mathbf{z}^{ij,l} \leftarrow \mathbb{1}\{ij \in \mathcal{E}_l\}\left(\bar{\mathbf{V}}^{ij,l}\right)_{\ell=0},\ ij \in \mathcal{E}$
  $(\mathbf{x}^{ij,l}, w^{ij,l}) \leftarrow \mathrm{MLP}_{\mathrm{latent}_l}\left(\left[\mathbf{x}^{ij,0}, \ldots, \mathbf{x}^{ij,l-1}, \mathbf{z}^{ij,l}\right]\right)$
return $E \leftarrow \sum_{ij\in\mathcal{E}} \mathrm{MLP}_{\mathrm{out}}\left[\mathbf{x}^{ij,0}, \mathbf{x}^{ij,1}, \ldots, \mathbf{x}^{ij,L}\right]$

Figure 2: (a) Schematic of the Branchlegro network. Blue and red arrows represent scalar and tensor features. Z_i is the atomic species of atom i, $\mathbf{r}_{ij}$ is the relative displacement vector from atom i to j. For simplicity, the residual update is not presented. Continuation gates modulate the tensor features and they are routed out in inference when they have a zero-gate (dashed red arrows). Routed scalar features are concatenated back for the energy readout (dashed blue arrows). (b) Pseudocode of the Branchlegro forward pass. For edge ij and layer l, $\mathbf{x}^{ij,l}$ are the scalar features, $w^{ij,l}$ are the environment weights, $\bar{\mathbf{V}}^{ij,l}$ are the tensor features, $\mathbf{Y}^{ij}$ are the spherical harmonic embeddings, L is the number of layers, and $\mathcal{F}$ is a gating function. The initial embedding is the standard Allegro one.

results in non-conservative dynamics. **Adaptive-precision potentials** use a continuous switching parameter, computed from hand-crafted descriptors, to mix the energies of fast and accurate potentials (Immel et al., 2025). The **Universal Models for Atoms** architecture uses a mixture-of-linear-experts framework to scale FMs without increasing inference cost (Wood et al., 2026). A router predicts expert weights from global time-invariant system properties, producing a single, fixed routing assignment during simulations. Liu et al. (2026) modified this to element-wise routing. **Static multifidelity Allegro** assigns models to fixed, predefined regions based on chemical intuition (Nascimento et al., 2026). Han et al. (2026) introduce a **dynamic cutoff** for atoms using a fixed function. Because BranchIP adapts computation within a single model, it is complementary to these architectures.

Early exits, introduced by Teerapittayanon et al. (2016), adapt NN inference to input complexity by attaching intermediate prediction heads. Sufficiently simple inputs can terminate before the final layer, reducing average compute while retaining full depth for harder inputs. Early exits were initially developed for convolutional NNs (Teerapittayanon et al., 2016; Huang et al., 2018) and later extended to transformers (Xin et al., 2020) and graph NNs (GNNs) (Di Francesco et al., 2026). **Learned adaptive computation** has also been explored at finer levels of granularity, including dynamically skipping network blocks (Wang et al., 2018; Wu et al., 2018), selectively evaluating feature channels (Gao et al., 2019; Hua et al., 2019), and varying message-passing computation across nodes in GNNs (Han et al., 2024; Abbahaddou et al., 2025). These approaches show that computation can also be allocated dynamically across intermediate components. To the best of our knowledge, learned dynamic allocation of equivariant computation across model depth has not been explored.

3 LEARNING ADAPTIVE EQUIVARIANT COMPUTATION

BranchIP augments an MLIP with lightweight routers placed before selected TP layers. At layer l, for each feature k (node or edge, depending on the architecture), a router produces an invariant

continuation gate, $f^{k,l} \in [0, 1]$, that modulates the feature's downstream contribution. The router may be learned or heuristic-based and may use any local information, including intermediate embeddings. At inference, zero-gated features are omitted from subsequent TPs, reducing compute and memory usage; we refer to this as sparse evaluation. During training, zero-gated features are retained to preserve supervision for downstream routers; we refer to this as dense evaluation. Therefore, BranchIP allocates depth adaptively across features, rather than applying the full depth uniformly.

Because conservative MLIPs obtain forces from energy derivatives, dynamic routing must remain part of the differentiable network computation. BranchIP therefore multiplies each feature by its continuation gate before the next TP. The gate makes routing a continuous operation on the features rather than a discrete selection, such that a feature's contribution decays smoothly to zero as it is routed out. To preserve the regularity of the PES, the continuation gate must be continuously differentiable to the same order as the MLIP it augments, with a flat exact-zero region. The flat region ensures that a feature reaches zero smoothly enough that its derivatives also vanish; as a result it carries no residual contribution when it is terminated. Zero-gated features can therefore be omitted from subsequent TP computation without changing the energy or its derivatives.

This design ensures that the sparse inference path reproduces the dense evaluation exactly, such that the routed model inherits the smoothness and symmetry guarantees of the architecture it augments.

Theorem 1. *Let an $E(3)$-equivariant twice continuously differentiable (C^2) MLIP be augmented with routers such that: (i) the continuation gate is $E(3)$-invariant, C^2 with a flat exact-zero region, and the same gating function is applied to every feature.; (ii) features are removed from TP computation only where the gate is exactly zero. Then, for all atomic coordinates $\mathbf{R}$:*

1. Regularity. *The PES is C^2 in $\mathbf{R}$, and the sparse computation yields its exact first and second derivatives. In particular, forces and stresses obtained from the sparse computation are exact.*
2. Symmetry. *$E(3)$-equivariance of the tensor features, invariance of the energy, and permutation symmetry are preserved.*

Proof intuition. The argument rests on the equivalence between the dense and sparse evaluations. The TP vanishes when its gated input is zero. Thus, by (ii), removed features do not contribute to the TP and both evaluations produce the same energy. For derivatives (e.g., forces), the sparse evaluation never builds backward computation branches for removed features. Thus, any nonzero derivative of the gate at a removed feature would contribute to the dense evaluation but be absent from the sparse evaluation. To prevent this, (i) requires the gate to be identically zero on an interval before a feature is removed, and to be C^2. Its derivatives therefore vanish on the interval and, by continuity, at the boundary, so each removed feature has zero value, zero slope, and zero curvature. Removing zero-gated features leaves the energy and its derivatives unchanged, so the sparse path computes the exact gradient of the same C^2 energy surface as the dense path. Finally, by the invariance and identical application in (i), the active feature set is unchanged under rotation and is permuted consistently under relabeling, so routing preserves the symmetries of the model. The same argument holds at any order: a C^m gate yields a C^m surface when the underlying MLIP is C^m.

For learned routing, we propose lightweight MLP routers trained by distilling a frozen full-compute teacher MLIP, from which the student BranchIP model is initialized. In the MLPs we use a sigmoid activation function and interpret the output as continuation probabilities, $\nu^{k,l} \in (0, 1)$. As a sigmoid cannot reach exact zeros, we define the gate as $f^{k,l} = \mathcal{F}(\nu^{k,l})$, where $\mathcal{F}$ satisfies the properties above. To train the router MLPs, we propose the following distillation loss, inspired by binary cross-entropy. We define a student-teacher prediction discrepancy, $\delta^k = \left\|\mathbf{y}^k_{\text{teacher}} - \mathbf{y}^k_{\text{student}}\right\|_2$ on a chosen output $\mathbf{y}^k$ (e.g., energy, forces). A low δ^k indicates that further TP computation is unnecessary, whereas a high δ^k indicates that computation should continue. User-defined stop and continue tolerances, τ^l_{cont} and τ^l_{stop}, convert δ^k into learnable router targets optimized with the distillation loss,

$$\mathcal{L}_{\text{distill}} = -\sum_{l=1}^{L} \sum_{k \in \mathcal{K}_l} r^{k,l} \left[\mathbb{1}_{\{\delta^k > \tau^l_{\text{stop}}\}} \log \nu^{k,l} + \mathbb{1}_{\{\delta^k \leq \tau^l_{\text{cont}}\}} \log\big(1 - \nu^{k,l}\big) \right], \tag{1}$$

where $\mathcal{K}_l$ is the set of active features in layer l and $r^{k,l} = \text{sg}[\prod_{m<l} \nu^{k,m}]$ weights each term by the fraction of feature k's signal that reaches router l. sg denotes the stop-gradients operator. This prevents earlier routers from reducing downstream distillation loss by decreasing their continuation

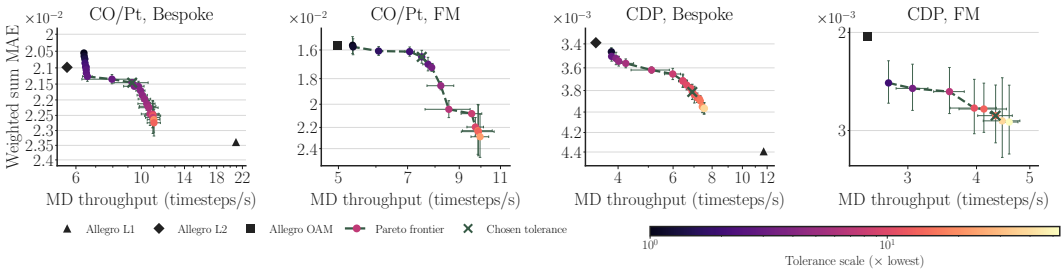


Figure 3: Accuracy–throughput trade-offs for Branchlegro on CO/Pt and CDP using bespoke models and FMs. Each point reports the mean test weighted-sum MAE and MD throughput over seven seeds; error bars denote standard deviation. The inverted y-axis indicates higher accuracy (higher is better). Weights for CO/Pt are 1:1:1 for per-atom energy, forces, and stress; for CDP are 10:1 for per-atom energy and forces. Throughput was measured as described in Appendix D using system sizes of 4,160 and 512 atoms for CO/Pt bespoke models and FMs, and 14,336 and 768 atoms for CDP bespoke models and FMs. Black markers denote Allegro baselines. Only Pareto frontier points are shown; tolerance scaling stops once no new frontier points are found. The $\times$ markers indicate the configurations used for the main results.

probabilities. The loss penalizes unnecessary continuation when the discrepancy falls below the continue tolerance and premature stopping when it exceeds the stop tolerance. The tolerances should obey $\tau_{\text{cont}} \leq \tau_{\text{stop}}$ and discrepancies between the two tolerances are left unsupervised, providing a transition region between continuation and stopping.

Branchlegro. We implement BranchIP in Allegro, denoted Branchlegro. In the implementation, MLPs route features that represent edges between atoms. We train the routers using the proposed distillation loss with force predictions (see Appendix A for formulation and gating function). Only the TP is sparsified, with terminated edges continuing to receive scalar latent updates. The skipped TP outputs are zero-padded to the full edge set and environment weights are evaluated over all edges at every layer. Allegro's radial envelope vanishes to second order at the cutoff, so the architecture is C^2. We therefore use a C^2 gate, identically zero below a threshold, matching that regularity. A smoother gate is available by increasing p in the continuation function. The Branchlegro architecture schematic and pseudocode are presented in Figure 2. Branchlegro therefore satisfies (i) and (ii), and Theorem 1 follows from Propositions 1–3 and Corollary 1, proved in Appendix B.

4 Results

We trained Branchlegro models from bespoke Allegro potentials and fine-tuned Allegro-OAM-L FMs (Kavanagh et al., 2026) by optimizing a joint objective comprising the proposed distillation loss and standard energy, force, and, for CO/Pt, stress supervision using seven random seeds (Appendix C). We evaluate each Branchlegro model against the Allegro teacher it was distilled from: two-layer (L2) models for bespoke potentials and the fine-tuned Allegro-OAM-L for the FMs. For the bespoke models, we also include one-layer (L1) Allegro as an efficiency comparison at lower accuracy, whereas no lower-compute variant is available for Allegro-OAM-L.

We examine the trade-off between predictive accuracy and MD throughput in Branchlegro by increasing the routing tolerances (Figure 3). Across all four settings, relaxing the tolerances yields substantial throughput gains with only moderate increases in error up to a model-dependent knee. Some tolerance profiles fall near transitions between routing regimes, where small differences in optimization can change which routers are favored, increasing the variance in throughput across seeds. For the CDP FM models, two of seven seeds result in worse validation and test errors across all tolerance profiles leading to the larger error variance. We select models near the corresponding validation-set knees for the system-specific scaling and physical studies that follow. These serve as representative models; in practice, Branchlegro routing tolerances can be tuned to match the accuracy and throughput requirements of a given application.

Speedup and memory. Figure 4 shows the MD throughput speedup and memory usage of Branchlegro over Allegro (Appendix D). In our experiments, Branchlegro models achieve 1.6–2.4$\times$ speedup

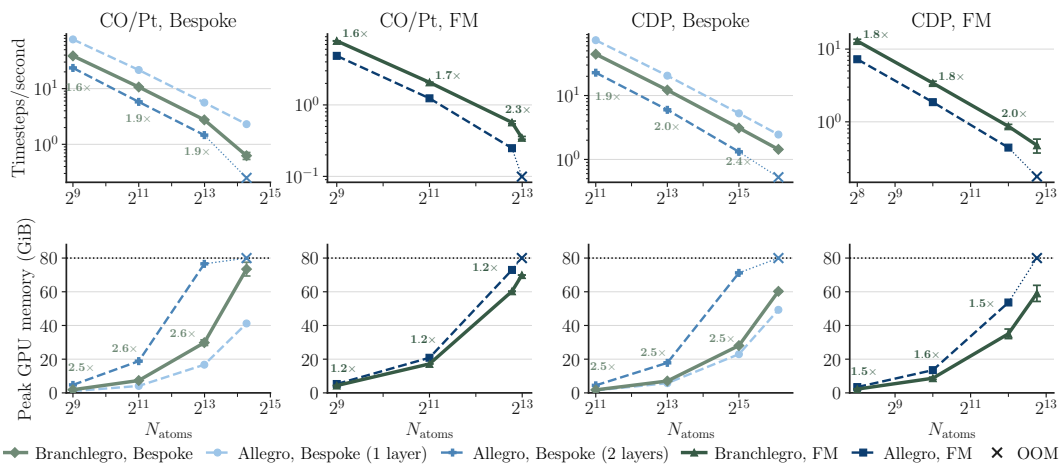


Figure 4: MD throughput and memory usage scaling of Allegro and Branchlegro on CO/Pt and CDP. Branchlegro values are averaged over seven seeds, with error bars indicating the standard deviation. All evaluations followed the setup in Appendix D. Speedups and memory reductions are relative to the corresponding full-sized Allegro model. Crosses denote out-of-memory (OOM).

across model sizes and systems. Notably, accelerations emerge even at small scale: the ∼100K-parameter bespoke Branchlegro CDP model is $1.9\times$ faster with 2,048 atoms. Branchlegro also reduces memory requirements, enabling inference regimes inaccessible to Allegro by reducing memory usage by 1.2–$2.6\times$. The FMs may achieve smaller memory reductions due to their significantly larger hidden dimension and latent MLPs. The bespoke models achieve significant memory reductions even in the regime of smaller speedups (e.g., $2.5\times$ memory reduction for CO/Pt bespoke on 512 atoms, with only a $1.6\times$ speedup). The speedup and memory reductions reported reflect the selected models and systems and not an inherent upper limit of Branchlegro performance. For example, by choosing a CO/Pt FM Branchlegro model with a larger tolerance scale, the speedup increases to $3.2\times$ and memory reduction to $1.8\times$, at the cost of further accuracy degradation (Figure 3).

Accuracy. We evaluate per-atom energy, force, and stress mean absolute errors (MAEs) (Table 1), alongside physical fidelity to ground-truth calculations (Appendix E). Branchlegro maintains comparable MAEs while retaining energy profiles and structural and dynamical observables (Appendix E). Notably, bespoke Branchlegro models achieve this accuracy despite most edges stopping before the first TP layer. For CO/Pt, the L1 Allegro model exhibits a lower degradation in errors relative to the L2 model than in CDP. This suggests that CO/Pt requires fewer interaction layers, consistent with the lower errors achieved by Branchlegro. Branchlegro models obey energy conservation in MD (Appendix F). We further demonstrate the practicality of Branchlegro models in realistic MLIP applications, large length- and time-scale MD. The CDP MLIP provides explicit simulation of long-range proton transport from which observables such as proton-diffusion activation energy can be determined. Branchlegro yields a proton-diffusion activation energy of $E_a = 0.40$ eV, within the

Table 1: Accuracy (MAE) of Allegro and Branchlegro on test datasets. Branchlegro errors are shown as mean ± standard deviation over seven seeds. Superscript $\hookleftarrow$ denotes a router before a layer.

	CO/Pt			CDP	
Architecture$_{\{\text{Layers}\}}$	E [meV/atom]	F [meV/Å]	S [meV/Å^3]	E [meV/atom]	F [meV/Å]
Allegro$_{\{0\}}$	2.50	67.4	0.298	0.805	40.3
Allegro$_{\{0,1\}}$	2.53	60.2	0.267	0.580	31.5
Branchlegro$_{\{\hookleftarrow 0,\hookleftarrow 1\}}$	2.16 ±0.104	61.9 ±0.216	0.277 ±0.00554	0.628 ±0.0260	35.7 ±0.682
Allegro$_{\{0,1,2,3\}}$	1.56	45.4	0.234	0.339	19.0
Branchlegro$_{\{\hookleftarrow 0,\hookleftarrow 1,\hookleftarrow 2,\hookleftarrow 3\}}$	1.77 ±0.298	47.3 ±1.22	0.266 ±0.00791	0.441 ±0.0571	26.6 ±3.17

experimental range of 0.39–0.43 eV (Haile et al., 2007; Ishikawa et al., 2008) and close to its L2 teacher, whereas L1 Allegro is outside the experimental range with $E_a = 0.46$ eV.

4.1 INTERPRETABILITY

The learned routing provides model interpretability by showing which atomic environments require deeper equivariant computation. We analyze the routing patterns of the two systems and focus on the four routers in the Branchlegro FMs, as they provide a richer analysis than the two routers of the bespoke models.

CO/Pt. The system consists of a Pt slab interacting with CO gas molecules, where carbon atoms adsorb to the surface. The system contains a relatively regular Pt bulk together with a chemically heterogeneous and complex surface. On the surface, Pt–C adsorption, C–O bonding, lateral CO interactions, and local surface Pt geometry are strongly coupled (Owen et al., 2024). The routers retain mostly edges involving carbon and oxygen and a small subset of the Pt–Pt edges (Figure 5a). Notably, this routing tracks the chemistry of the reactive surface: the fourth router retains adsorbed CO while routing out most gas-phase CO, and the second and third routers preferentially retain edges of Pt atoms with adsorbed CO relative to those without (Figure 5c). We analyze these patterns as a function of edge distance in Figure 5d for Pt–C/O edges (centered on Pt) and for Pt–Pt edges.

Pt–C and Pt–O edges are retained in the final layer at around 2 Å and 3 Å, the expected distances of adsorbed C (Ogletree et al., 1986) and its O atom. In contrast, Pt edges with gaseous CO are terminated by the final router. For Pt–Pt edges, we classify the central Pt atom as bulk, bare surface, or adsorbate-bound in a linear single coordination (top) or multiple coordination (multi) configuration (see Appendix G). Strikingly, the second router differentiates bulk, bare surface, and adsorbate-bound Pt atoms, while the third router further separates the two adsorbate configurations. Since Pt–Pt separations are largely unchanged by adsorption, this pattern indicates that the routers identify these environments as more complex, requiring higher body-order information, and allocate additional TP refinement to them accordingly.

CDP. The system is composed of a relatively localized Cs and P framework, orientationally disordered PO_4 tetrahedra, and highly mobile protons (H). The framework edges (Cs–Cs, Cs–P, and P–P) are mainly retained by the earlier routers, but are entirely terminated by the final router (Figure 5b). Conversely, a substantial portion of edges involving H or O atoms are terminated in early routers, while a small subset is retained through the final router. This suggests these interactions require deeper many-body computation. We inspect the routing patterns over edge distance in Figure 5d for H–P edges (centered on H) and for O–O edges.

For H–P, the edges are classified by the proton's instantaneous bonding relationship to the PO_4 group: covalent, when the proton is covalently bonded to one of its oxygens; hydrogen-bonded, when the proton forms a hydrogen bond to one of the oxygens in the same PO_4 group (H$\cdots$O–P); and unrelated, when neither holds. The classes are well separated in distance, with the covalent and hydrogen-bonded classes peaking near 2.3 and 3.0 Å and the unrelated class beginning beyond 4 Å. As the proton rattles on sub-picosecond timescales between the two oxygens of a hydrogen bond (Wang et al., 2025), which belong to different PO_4 groups, a given phosphate pair exchanges the two labels repeatedly along a trajectory. In the Branchlegro model, the first router terminates nearly all unrelated H–P edges, the third terminates the covalent edges, and the fourth terminates the hydrogen-bonded edges. The early removal of unrelated edges indicates that distant phosphate environments contribute little once the local proton environment has been established. The two bonded classes differ in how much of the H–P geometry the covalent bonding already fixes. For the covalent class, the near-rigid P–O–H geometry fixes it (Wang et al., 2025), accounting for the first peak. The H–P edge therefore adds little beyond the covalent H–O edge already present in the same proton-centered environment. For the hydrogen-bonded class, no covalent bond constrains the H–P geometry. The geometry instead depends on how two reorienting PO_4 groups are arranged relative to one another, and thus responds to the rotation of either. These edges therefore remain for deeper many-body interactions.

For O–O, the edges are classified by the bonding relationship between the two oxygens: same-PO_4, for an O–O pair within one group; O-pair, for an O–H$\cdots$O hydrogen bond between two groups that share a proton (Wang et al., 2025); and other, for the remaining pairs in different groups. The same-PO_4 edges terminate at the second router, while the O-pair edges are retained at full amplitude

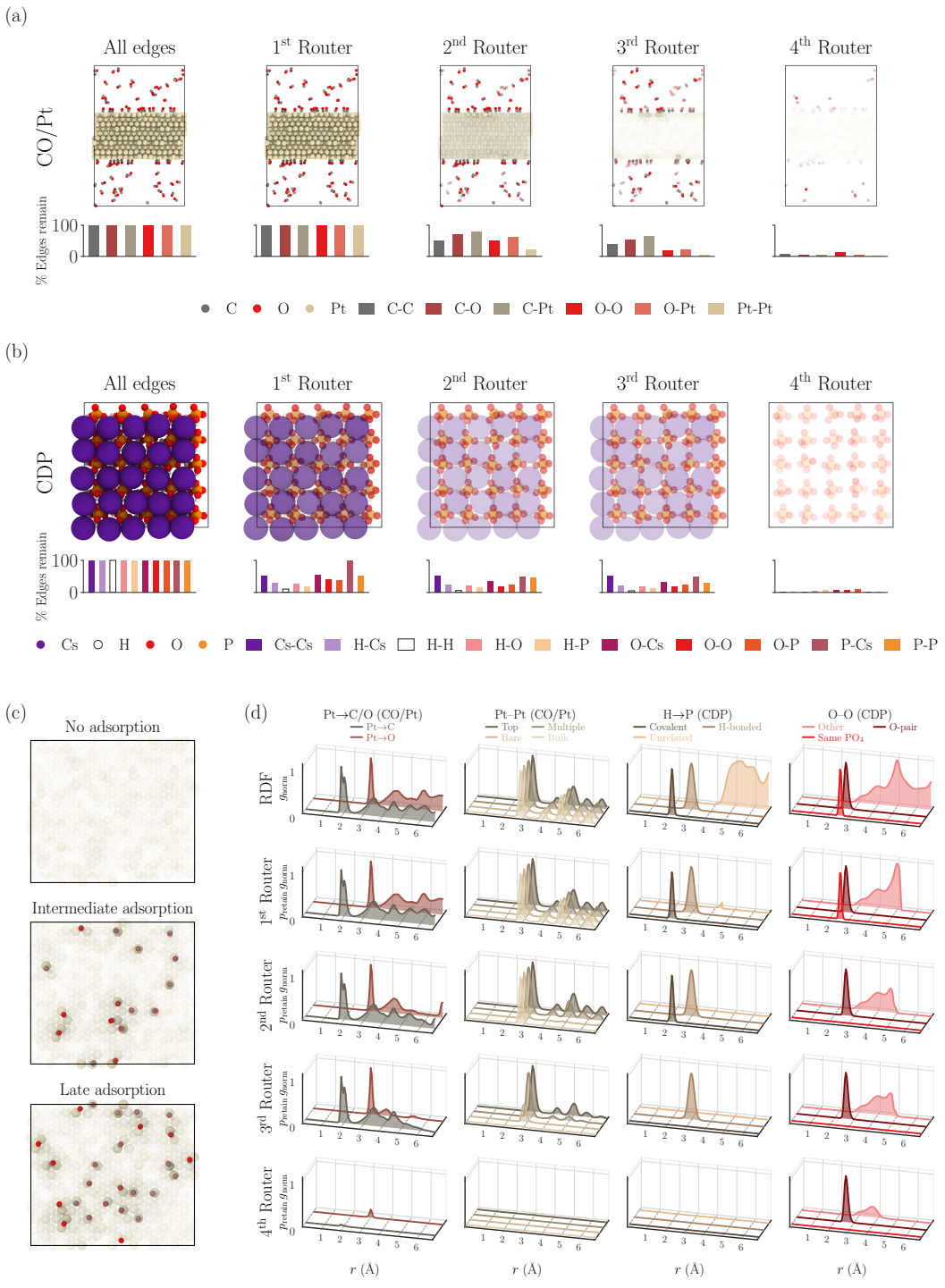


Figure 5: Adaptive routing in CO/Pt and CDP Branchlegro FMs. Atom opacity represents the cumulative gating of its connected edges retained in the TP operations after each router. (a) and (b) CO/Pt and CDP structures before routing and after Routers 1–4. For clarity, the CO/Pt visualization retains only the two closest planes of Pt atoms. (c) Top views of CO/Pt at Router 3 for configurations with varying CO adsorption. Only the upper two Pt layers and CO molecules within 2.2 Å of the uppermost Pt layer are shown. (d) Unweighted normalized radial distribution functions (RDFs), g_{norm}, and retention-weighted RDFs, $p_{\text{retain}} g_{\text{norm}}$, after Routers 1–4 for Pt–C/O and Pt–Pt interactions in CO/Pt and H–P and O–O in CDP. RDFs are normalized once using the initial peak value.

through all four routers. Because the two classes nearly coincide in distance, this separation cannot come from radial distance alone. At two-body order (Router 1), the router sees only species and distance, and accordingly retains both classes. The first TP adds, at each oxygen, the angle between the edge and that oxygen's own P–O bond, which identifies the pair as intramolecular. The edge is then redundant, since the P–O edges already encode the tetrahedron's orientation, and the second router terminates the edge. In contrast, an O-pair edge depends on the position of the proton between the two oxygens. That position is continually modified by rattling within the O-pair and by O-sharing, in which two protons from different O-pairs simultaneously share a single oxygen (Wang et al., 2025). Resolving this O-pair edge requires a larger neighborhood than in the intramolecular case, and therefore features of higher many-body order, consistent with the model retaining these edges through the final router.

5 Conclusion

We introduce BranchIP, a learned, dynamic adaptive inference framework for equivariant MLIPs that obeys energy conservation. The framework leverages the insight that features representing certain atomic environments have sufficient TP refinement and can avoid additional computation in later layers. Through our development of a novel distillation loss, learned routers provide model interpretability and accelerate inference without relying on chemical intuition. Our implementation, Branchlegro, in the MLIP Allegro achieves significant inference speedups and memory efficiency on complex reactive systems: heterogeneous CO/Pt and CDP, a system with no clear spatial partition. Our results show that adaptive computation creates faster MLIPs without hand-crafted routing, exorbitant parameter counts, or loss of physical fidelity.

Limitations. The Branchlegro implementation only reduces the TP computations, while the rest of the computations done in the layer are retained as in a standard Allegro model. This limits the speedup potential of Branchlegro, especially when there are other expensive operations (e.g., large edge-wise MLPs). In addition, the learned routing scheme with the novel distillation loss requires a pretrained teacher model, increasing the cost of the overall training. The experiments included in this work focus on two systems.

AI use statement

In this work, we used generative AI tools for assisting in the writing of proofs, providing feedback on research methodology or experiments, and implementing methods. We have not used generative AI tools for helping to develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, proposing or refining hypotheses, designing research methodology or experiments, cleaning and reformatting dataset, supporting qualitative and thematic data analysis, or interpreting results, and generating synthetic data sets and assisting with translation are not applicable to this work. Additionally, we used generative AI tools for refining figures and polishing of the manuscript. We have reviewed all AI-assisted work. Multiple authors verified and tested for correctness LLM-generated code, text, and proof writing suggestions. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility statement

Training setup and hyperparameters for all settings are reported in Appendix C. Details on throughput and memory measurements are reported in Appendix D. All BranchIP code used in this work will be made available in the OpenReview discussion.

References

Yassine Abbahaddou, Fragkiskos D. Malliaros, Johannes F. Lutzeyer, and Michalis Vazirgiannis. ADMP-GNN: Adaptive Depth Message Passing GNN, September 2025. URL http://arxiv.org/abs/2509.01170. arXiv:2509.01170 [cs.LG].

Ilyes Batatia, David P Kovacs, Gregor Simm, Christoph Ortner, and Gabor Csanyi. MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields. In S. Koyejo,

S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 11423–11436. Curran Associates, Inc., 2022. doi: 10.52202/068431-0830. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/4a36c3c51af11ed9f34615b81edb5bbc-Paper-Conference.pdf.

Ilyes Batatia, Philipp Benner, Yuan Chiang, Alin M. Elena, Dávid P. Kovács, Janosh Riebesell, Xavier R. Advincula, Mark Asta, Matthew Avaylon, William J. Baldwin, Fabian Berger, Noam Bernstein, Arghya Bhowmik, Filippo Bigi, Samuel M. Blau, Vlad Cărare, Michele Ceriotti, Sanggyu Chong, James P. Darby, Sandip De, Flaviano Della Pia, Volker L. Deringer, Rokas Elijošius, Zakariya El-Machachi, Edvin Fako, Fabio Falcioni, Andrea C. Ferrari, John L. A. Gardner, Mikołaj J. Gawkowski, Annalena Genreith-Schriever, Janine George, Rhys E. A. Goodall, Jonas Grandel, Clare P. Grey, Petr Grigorev, Shuang Han, Will Handley, Hendrik H. Heenen, Kersti Hermansson, Cheuk Hin Ho, Stephan Hofmann, Christian Holm, Jad Jaafar, Konstantin S. Jakob, Hyunwook Jung, Venkat Kapil, Aaron D. Kaplan, Nima Karimitari, James R. Kermode, Panagiotis Kourtis, Namu Kroupa, Jolla Kullgren, Matthew C. Kuner, Domantas Kuryla, Guoda Liepuoniute, Chen Lin, Johannes T. Margraf, Ioan-Bogdan Magdău, Angelos Michaelides, J. Harry Moore, Aakash A. Naik, Samuel P. Niblett, Sam Walton Norwood, Niamh O'Neill, Christoph Ortner, Kristin A. Persson, Karsten Reuter, Andrew S. Rosen, Louise A. M. Rosset, Lars L. Schaaf, Christoph Schran, Benjamin X. Shi, Eric Sivonxay, Tamás K. Stenczel, Christopher Sutton, Viktor Svahn, Thomas D. Swinburne, Jules Tilly, Cas van der Oord, Santiago Vargas, Eszter Varga-Umbrich, Tejs Vegge, Martin Vondrák, Yangshuai Wang, William C. Witt, Thomas Wolf, Fabian Zills, and Gábor Csányi. A foundation model for atomistic materials chemistry. *The Journal of Chemical Physics*, 163(18):184110, November 2025. ISSN 0021-9606. doi: 10.1063/5.0297006. URL https://doi.org/10.1063/5.0297006.

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E. Smidt, and Boris Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13(1):2453, May 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-29939-5. URL https://www.nature.com/articles/s41467-022-29939-5.

Jörg Behler and Michele Parrinello. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Physical Review Letters*, 98(14):146401, April 2007. ISSN 0031-9007, 1079-7114. doi: 10.1103/PhysRevLett.98.146401. URL https://link.aps.org/doi/10.1103/PhysRevLett.98.146401.

Vivek Bharadwaj, Austin Glover, Aydin Buluc, and James Demmel. An Efficient Sparse Kernel Generator for O(3)-Equivariant Deep Networks, May 2025. URL http://arxiv.org/abs/2501.13986. arXiv:2501.13986 [cs.LG].

Fraser Birks, Matthew Nutter, Thomas D. Swinburne, and James R. Kermode. Efficient and accurate spatial mixing of machine learned interatomic potentials for materials science. *npj Computational Materials*, 12(1):110, February 2026. ISSN 2057-3960. doi: 10.1038/s41524-026-01982-6. URL https://www.nature.com/articles/s41524-026-01982-6.

Chi Chen and Shyue Ping Ong. A universal graph deep learning interatomic potential for the periodic table. *Nature Computational Science*, 2(11):718–728, November 2022. ISSN 2662-8457. doi: 10.1038/s43588-022-00349-3. URL https://doi.org/10.1038/s43588-022-00349-3.

Beatriz Cordero, Verónica Gómez, Ana E. Platero-Prats, Marc Revés, Jorge Echeverría, Eduard Cremades, Flavia Barragán, and Santiago Alvarez. Covalent radii revisited. *Dalton Transactions*, 37(21):2832–2838, January 2008. ISSN 1477-9226. doi: 10.1039/b801115j. URL https://doi.org/10.1039/b801115j. _eprint: https://pubs.rsc.org/dt/article-pdf/37/21/2832/1579248/b801115j.pdf.

Isabel Creed, Tim Rein, Ingvars Vitenburgs, Wojciech G. Stark, Viktor Ellingsson, Ahmed Y. Ismail, Guangyu Liu, Yuchen Lou, Bradley A. A. Martin, Cyprien Bone, Matthew A. H. Walker, Mueen Taj, Shirui Wang, Kelvin Wong, Ruiqi Wu, Prakriti Kayastha, Bingqing Cheng, Aditi Krishnapriyan, Michele Ceriotti, Marcel F. Langer, Jarvist Moore Frost, Alex M. Ganose, Venkat Kapil, and Keith T. Butler. Six Open Questions in Machine-Learned Interatomic Potential Foundation Models, 2026. URL https://arxiv.org/abs/2606.07327. _eprint: 2606.07327.

Andrea Giuseppe Di Francesco, Maria Sofia Bucarelli, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, and Fabrizio Silvestri. Early-Exit Graph Neural Networks, February 2026. URL http://arxiv.org/abs/2505.18088. arXiv:2505.18088 [cs.LG].

Daan Frenkel and Berend Smit (eds.). *Understanding Molecular Simulation: From Algorithms to Applications*. Academic Press, Inc., USA, 1st edition, August 1996. ISBN 978-0-12-267370-2.

Xitong Gao, Yiren Zhao, Łukasz Dudziak, Robert Mullins, and Cheng-zhong Xu. Dynamic Channel Pruning: Feature Boosting and Suppression. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJxh2j0qYm.

Sossina M. Haile, Calum R. I. Chisholm, Kenji Sasaki, Dane A. Boysen, and Tetsuya Uda. Solid acid proton conductors: from laboratory curiosities to fuel cell electrolytes. *Faraday Discussions*, 134:17–39, January 2007. ISSN 1359-6640. doi: 10.1039/b604311a. URL https://doi.org/10.1039/b604311a. _eprint: https://pubs.rsc.org/fd/article-pdf/doi/10.1039/b604311a/1249343/b604311a.pdf.

Kevin Han, Haolin Cong, Bowen Deng, and Amir Barati Farimani. Smooth Dynamic Cutoffs for Machine Learning Interatomic Potentials. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=BCVB3TFpk5.

Yan Han, Kaiqi Chen, Shan Li, Ji Yan, Baoxu Shi, Lei Zhang, Fei Chen, Jaewon Yang, Yunpeng Xu, Xiaoqiang Luo, Qi He, Ying Ding, and Zhangyang Wang. Turning A Curse into A Blessing: Data-Aware Memory-Efficient Training of Graph Neural Networks by Dynamic Exiting. In *Companion Proceedings of the ACM Web Conference 2024*, pp. 903–906, Singapore Singapore, May 2024. ACM. ISBN 979-8-4007-0172-6. doi: 10.1145/3589335.3651575. URL https://dl.acm.org/doi/10.1145/3589335.3651575.

Gil Harari, Yoel Zimmermann, Ola Tangen Kulseng, Laura Zichi, Chuin Wei Tan, Marc L. Descoteaux, and Boris Kozinsky. Beyond Adam: SOAP and Muon for Faster, Label-Efficient Training of Machine Learning Interatomic Potentials, July 2026. URL http://arxiv.org/abs/2607.02499. arXiv:2607.02499 [cs.LG].

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training Compute-Optimal Large Language Models, March 2022. URL http://arxiv.org/abs/2203.15556. arXiv:2203.15556 [cs.CL].

Weizhe Hua, Yuan Zhou, Christopher De Sa, Zhiru Zhang, and G. Edward Suh. Channel gating neural networks. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens van der Maaten, and Kilian Weinberger. Multi-Scale Dense Networks for Resource Efficient Image Classification. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hk2aImxAb.

David Immel, Ralf Drautz, and Godehard Sutmann. Adaptive-precision potentials for large-scale atomistic simulations. *The Journal of Chemical Physics*, 162(11):114119, March 2025. ISSN 0021-9606. doi: 10.1063/5.0245877. URL https://doi.org/10.1063/5.0245877.

Ayumu Ishikawa, Hideki Maekawa, Tsutomu Yamamura, Yukinobu Kawakita, Kaoru Shibata, and Masayoshi Kawai. Proton dynamics of CsH2PO4 studied by quasi-elastic neutron scattering and PFG-NMR. *Solid State Ionics*, 179(40):2345–2349, 2008. ISSN 0167-2738. doi: https://doi.org/10.1016/j.ssi.2008.10.002. URL https://www.sciencedirect.com/science/article/pii/S0167273808005997.

Anders Johansson, Evan Weinberg, Christian Trott, Megan McCarthy, and Stan Moore. LAMMPS-KOKKOS: Performance Portable Molecular Dynamics Across Exascale Architectures. In *Proceedings of the SC '25 Workshops of the International Conference for High Performance Computing,*

Networking, Storage and Analysis, SC Workshops '25, pp. 1217–1232, New York, NY, USA, November 2025. Association for Computing Machinery. ISBN 979-8-4007-1871-7. doi: 10.1145/3731599.3767498. URL https://dl.acm.org/doi/10.1145/3731599.3767498.

Bhupalee Kalita, Hatice Gokcan, and Olexandr Isayev. Machine learning interatomic potentials at the centennial crossroads of quantum mechanics. *Nature Computational Science*, 5(12):1120–1132, December 2025. ISSN 2662-8457. doi: 10.1038/s43588-025-00930-6. URL https://doi.org/10.1038/s43588-025-00930-6.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling Laws for Neural Language Models, January 2020. URL http://arxiv.org/abs/2001.08361. arXiv:2001.08361 [cs.LG].

Seán R. Kavanagh, Chuin Wei Tan, Menghang Wang, Marc L. Descoteaux, Gabriel de Miranda Nascimento, Ulrik Unneberg, Laura Zichi, Francesco Libbi, Norma Rivano, Austin Glover, Vivek Bharadwaj, Anders Johansson, William C. Witt, Albert Musaelian, and Boris Kozinsky. Fast and Accurate Foundation Models for Equivariant Machine-Learned Interatomic Potentials, July 2026. URL http://arxiv.org/abs/2607.28461. arXiv:2607.28461 [physics.comp-ph].

Boris Kozinsky, Albert Musaelian, Anders Johansson, and Simon Batzner. Scaling the Leading Accuracy of Deep Equivariant Models to Biomolecular Simulations of Realistic Size. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 979-8-4007-0109-2. doi: 10.1145/3581784.3627041. URL https://doi.org/10.1145/3581784.3627041.

Seung Yul Lee, Hojoon Kim, Yutack Park, Dawoon Jeong, Seungwu Han, Yeonhong Park, and Jae W. Lee. FlashTP: Fused, Sparsity-Aware Tensor Product for Machine Learning Interatomic Potentials. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 33143–33156. PMLR, July 2025. URL https://proceedings.mlr.press/v267/lee25l.html.

Yuzhi Liu, Duo Zhang, Anyang Peng, Weinan E, Linfeng Zhang, and Han Wang. Scaling Machine Learning Interatomic Potentials with Mixtures of Experts, March 2026. URL http://arxiv.org/abs/2603.07977. arXiv:2603.07977 [physics.chem-ph].

Shengjie Luo, Tianlang Chen, and Aditi Krishnapriyan. Enabling Efficient Equivariant Operations in the Fourier Basis via Gaunt Tensor Products. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 24742–24777, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/6b61c278e483954fee502b49fe71cd14-Paper-Conference.pdf.

Albert Musaelian, Simon Batzner, Anders Johansson, Lixin Sun, Cameron J. Owen, Mordechai Kornbluth, and Boris Kozinsky. Learning local equivariant representations for large-scale atomistic dynamics. *Nature Communications*, 14(1):579, February 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-36329-y. URL https://www.nature.com/articles/s41467-023-36329-y.

Gabriel de Miranda Nascimento, Marc L. Descoteaux, Laura Zichi, Chuin Wei Tan, William C. Witt, Nicola Molinari, Sriteja Mantha, Daniil Kitchaev, Mordechai Kornbluth, Karim Gadelrab, Charles Tuffile, and Boris Kozinsky. Mixture of Experts Framework in Machine Learning Interatomic Potentials for Atomistic Simulations, April 2026. URL http://arxiv.org/abs/2604.26143. arXiv:2604.26143 [physics.comp-ph].

D. F. Ogletree, M. A. Van Hove, and G. A. Somorjai. LEED intensity analysis of the structures of clean Pt (111) and of CO adsorbed on Pt (111) in the c (4\times 2) arrangement. *Surface science*, 173(2-3):351–365, 1986. URL https://www.sciencedirect.com/science/article/pii/0039602886901950?casa_token=CH3duuu47d0AAAAA:TOdplRZCBsPIv90hxNXGy_SCPSjqDL9ix0-2rva9LGY0H4mrwH76J0sHyZSbOJ8VC81L2L7N.

Cameron J. Owen, Nicholas Marcella, Christopher R. O'Connor, Taek-Seung Kim, Ryuichi Shimogawa, Clare Yijia Xie, Ralph G. Nuzzo, Anatoly I. Frenkel, Christian Reece, and Boris Kozinsky. Surface roughening in nanoparticle catalysts, July 2024. URL http://arxiv.org/abs/2407.13643. arXiv:2407.13643 [cond-mat.mtrl-sci].

Hillary Pan, Alex M. Ganose, Matthew Horton, Muratahan Aykol, Kristin A. Persson, Nils E. R. Zimmermann, and Anubhav Jain. Benchmarking Coordination Number Prediction Algorithms on Inorganic Crystal Structures. *Inorganic Chemistry*, 60(3):1590–1603, February 2021. ISSN 0020-1669. doi: 10.1021/acs.inorgchem.0c02996. URL https://doi.org/10.1021/acs.inorgchem.0c02996. _eprint: https://pubs.acs.org/inocaj/article-pdf/60/3/1590/10257641/ic0c02996.pdf.

Yutack Park, Jaesun Kim, Seungwoo Hwang, and Seungwu Han. Scalable Parallel Algorithm for Graph Neural Network Interatomic Potentials in Molecular Dynamics Simulations. *Journal of Chemical Theory and Computation*, 20(11):4857–4868, May 2024. ISSN 1549-9618. doi: 10.1021/acs.jctc.4c00190. URL https://doi.org/10.1021/acs.jctc.4c00190. _eprint: https://pubs.acs.org/jctcce/article-pdf/20/11/4857/1779290/ct4c00190.pdf.

Saro Passaro and C. Lawrence Zitnick. Reducing SO(3) Convolutions to SO(2) for Efficient Equivariant GNNs. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 27420–27438. PMLR, July 2023. URL https://proceedings.mlr.press/v202/passaro23a.html.

Chuin Wei Tan, Marc L. Descoteaux, Mit Kotak, Gabriel De Miranda Nascimento, Seán R. Kavanagh, Laura Zichi, Menghang Wang, Aadit Saluja, Yizhong R. Hu, Tess Smidt, Anders Johansson, William C. Witt, Boris Kozinsky, and Albert Musaelian. High-performance training and inference for deep equivariant interatomic potentials. *Digital Discovery*, 5(4):1558–1567, 2026. ISSN 2635-098X. doi: 10.1039/D5DD00423C. URL https://pubs.rsc.org/dd/article/5/4/1558-1567/1229667.

Surat Teerapittayanon, Bradley McDanel, and H.T. Kung. BranchyNet: Fast inference via early exiting from deep neural networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 2464–2469, Cancun, December 2016. IEEE. ISBN 978-1-5090-4847-2. doi: 10.1109/ICPR.2016.7900006. URL http://ieeexplore.ieee.org/document/7900006/.

Aidan P. Thompson, H. Metin Aktulga, Richard Berger, Dan S. Bolintineanu, W. Michael Brown, Paul S. Crozier, Pieter J. In 'T Veld, Axel Kohlmeyer, Stan G. Moore, Trung Dac Nguyen, Ray Shan, Mark J. Stevens, Julien Tranchida, Christian Trott, and Steven J. Plimpton. LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Computer Physics Communications*, 271:108171, February 2022. ISSN 00104655. doi: 10.1016/j.cpc.2021.108171. URL https://linkinghub.elsevier.com/retrieve/pii/S0010465521002836.

Menghang Wang, Jingxuan Ding, Grace Xiong, Ni Zhan, Cameron J. Owen, Albert Musaelian, Yu Xie, Nicola Molinari, Ryan P. Adams, Sossina Haile, and Boris Kozinsky. Revealing the proton slingshot mechanism in solid acid electrolytes through machine learning molecular dynamics, March 2025. URL http://arxiv.org/abs/2503.15389. arXiv:2503.15389 [cond-mat].

Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E. Gonzalez. SkipNet: Learning Dynamic Routing in Convolutional Networks. In *The European Conference on Computer Vision (ECCV)*, September 2018.

Brandon M. Wood, Misko Dzamba, Xiang Fu, Meng Gao, Muhammed Shuaibi, Luis Barroso-Luque, Kareem Abdelmaqsoud, Vahe Gharakhanyan, John R. Kitchin, Daniel S. Levine, Kyle Michel, Anuroop Sriram, Taco Cohen, Abhishek Das, Ammar Rizvi, Sushree Jagriti Sahoo, Zachary W. Ulissi, and C. Lawrence Zitnick. UMA: A Family of Universal Models for Atoms, March 2026. URL http://arxiv.org/abs/2506.23971. arXiv:2506.23971 [cs.LG].

Zuxuan Wu, Tushar Nagarajan, Abhishek Kumar, Steven Rennie, Larry S Davis, Kristen Grauman, and Rogerio Feris. BlockDrop: Dynamic Inference Paths in Residual Networks. In *CVPR*, 2018.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. DeeBERT: Dynamic Early Exiting for Accelerating BERT Inference, April 2020. URL http://arxiv.org/abs/2004.12993. arXiv:2004.12993 [cs.CL].

Zemin Xu, Chenyu Wu, Wenbo Xie, and Peijun Hu. A Cartesian-3j Framework for Machine Learning Interatomic Potentials. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=9ZWK6gneWq.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling Vision Transformers. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1204–1213, 2022. doi: 10.1109/CVPR52688.2022.01179.

Yong-Wei Zhang, Viacheslav Sorkin, Zachary H Aitken, Antonio Politano, Jörg Behler, Aidan P Thompson, Tsz Wai Ko, Shyue Ping Ong, Olga Chalykh, Dmitry Korogod, Evgeny Podryabinkin, Alexander Shapeev, Ju Li, Yuri Mishin, Zongrui Pei, Xianglin Liu, Jaesun Kim, Yutack Park, Seungwoo Hwang, Seungwu Han, Killian Sheriff, Yifan Cao, and Rodrigo Freitas. Roadmap for the development of machine learning-based interatomic potentials. *Modelling and Simulation in Materials Science and Engineering*, 33(2):023301, January 2025. ISSN 1361-651X. doi: 10.1088/1361-651x/ad9d63. URL http://dx.doi.org/10.1088/1361-651X/ad9d63.

A BRANCHLEGRO DETAILS

A.1 FORCE DISTILLATION LOSS FORMULATION

We detail the exact formulation for the force distillation loss component used to train our Branchlegro models. The features in Allegro are edge-based, so $\mathcal{K} = \mathcal{E}$. For the force discrepancy, we use the ℓ_2 norm of the force difference between the teacher and student predictions:

$$\delta_i^F = \left\| \mathbf{F}_i^{\text{teacher}} - \mathbf{F}_i^{\text{student}} \right\|_2, \tag{2}$$

where $\mathbf{F}_i^{\text{teacher/student}}$ is the force prediction of the teacher/student on atom i. As forces are node-resolved but the routers predict edge-wise probabilities, we convert the atomic force discrepancies to edge discrepancies:

$$\delta_{ij}^F = \frac{\delta_i^F + \delta_j^F}{2}. \tag{3}$$

Thus, for Branchlegro's edge features, Equation 1 becomes:

$$\mathcal{L}_{\text{distill}} = -\sum_{l=1}^{L} \sum_{ij \in \mathcal{E}_l} r^{ij,l} \left[\mathbb{1}_{\{\delta^{ij} > \tau_{\text{stop}}^l\}} \log \nu^{ij,l} + \mathbb{1}_{\{\delta^{ij} \leq \tau_{\text{cont}}^l\}} \log\left(1 - \nu^{ij,l}\right) \right], \tag{4}$$

where $\nu^{ij,l} = \text{MLP}_{\text{router}_l}(\mathbf{x}^{ij,l})$ and $r^{ij,l} = \text{sg}[\prod_{m<l} \nu^{ij,m}]$.

A.2 CONTINUATION-GATING FUNCTION

For the smooth continuation-gating function, we use

$$\mathcal{F}(x) = 1 - \exp\left[-\left(\left(\frac{x - T}{w} \right)_+ \right)^p \right] \tag{5}$$

where $(a)_+ \coloneqq \max\{a, 0\}$. For $p \in \mathbb{N}$ and $w > 0$, this construction ensures that $\mathcal{F}(x) \in [0, 1)$, $\mathcal{F} \in C^{p-1}$, and $\mathcal{F}(x) = 0$ for all $x \leq T$. Thus, we choose $p = 3$ to ensure the resulting PES is C^2. In our implementation, we use $T = 0.3$ and $w = 0.1$ for all models. These parameters provide coverage of the nearly ungated ($\mathcal{F}(x) \approx 1$) and fully gated ($\mathcal{F}(x) = 0$) regimes, with a sharp but smooth transition controlled by w. Empirically, the choice of T has little effect on the resulting routing patterns, suggesting that the router learns to compensate for changes in this threshold. Figure 6 illustrates the gating function and its dependence on these parameters.

We note the continuation-gating function is similar to the polynomial envelope function used in the `nequip`-framework to smoothly transition edge activations to zero at the radial cutoff of the model.

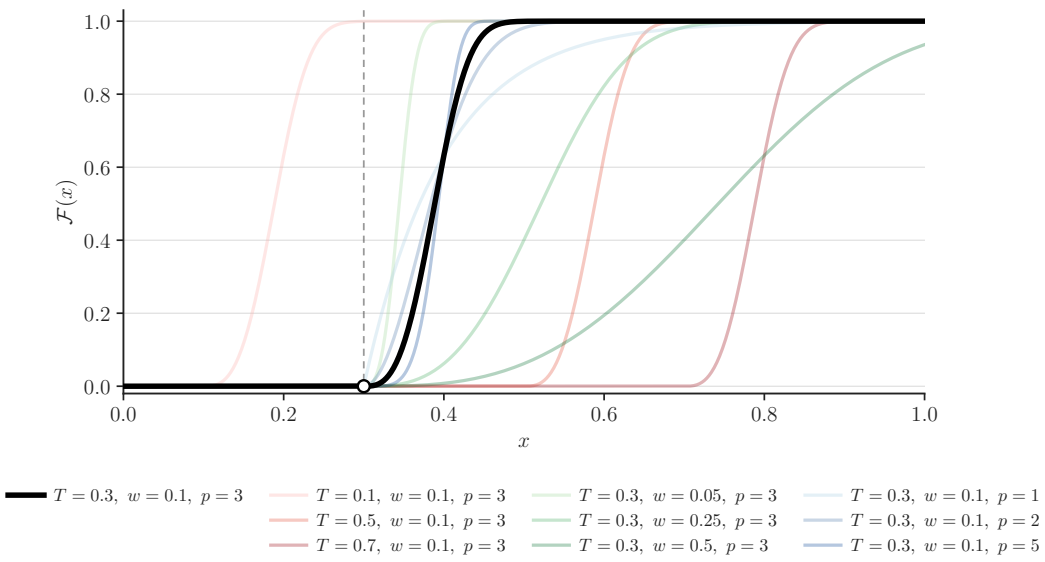


Figure 6: The continuation gate function used in Branchlegro. The black curve denotes the parameter setting used in this work.

B THEORETICAL GUARANTEES

Equivariant and energy-conserving MLIPs are constructed from operations that preserve their underlying symmetries and conservation laws. For conservative MLIPs, forces are obtained as derivatives of the predicted energy, so dynamic routing must not introduce discontinuities or remove nonzero derivative contributions. Likewise, operations acting on equivariant features must respect their transformation laws to preserve model equivariance.

BranchIP introduces dynamic, feature-level routing into an equivariant MLIP, altering which features participate in TP computation during inference. For this adaptive computation to preserve the model's physical properties, sparse evaluation must reproduce the corresponding dense gated energy and its derivatives, while preserving the model's equivariance and permutation symmetries. We establish these guarantees for Branchlegro.

Let $\mathbf{R}$ denote the atomic coordinates and let $\mathcal{E}$ be the full directed edge set constructed from the neighbor graph. At layer l, each edge $ij \in \mathcal{E}$ has scalar features $\mathbf{x}^{ij,l}$, environment weights $w^{ij,l}$, and equivariant tensor features $\bar{\mathbf{V}}^{ij,l}$, consistent with the Allegro forward pass. For clarity, we suppress irreducible representations and channel indices as well as environment tensor normalization, which do not affect the routing arguments below.

Define the environment tensor entering layer $l+1$ as

$$\mathbf{A}^{i,l} = \sum_{k:(i,k)\in\mathcal{E}} w^{ik,l}\mathbf{Y}^{ik}, \tag{6}$$

where $\mathbf{Y}^{ik}$ denotes the spherical harmonic embedding. The TP update is then

$$\bar{\mathbf{V}}^{ij,l+1} = \mathbf{A}^{i,l} \otimes \bar{\mathbf{V}}^{ij,l}. \tag{7}$$

The scalar component of the TP output is

$$\mathbf{z}^{ij,l+1} = \left(\bar{\mathbf{V}}^{ij,l+1}\right)_{\ell=0}, \tag{8}$$

where ℓ denotes angular momentum. The latent update is then

$$\left(\mathbf{x}^{ij,l+1}, w^{ij,l+1}\right) = \mathrm{MLP}_{\mathrm{latent}_{l+1}}\left(\left[\mathbf{x}^{ij,0}, \ldots, \mathbf{x}^{ij,l}, \mathbf{z}^{ij,l+1}\right]\right), \qquad ij \in \mathcal{E}. \tag{9}$$

The final energy is computed from the concatenated scalar latent features,

$$E(\mathbf{R}) = \sum_{ij\in\mathcal{E}} \mathrm{MLP}_{\mathrm{out}}\left(\left[\mathbf{x}^{ij,0}, \ldots, \mathbf{x}^{ij,L}\right]\right). \tag{10}$$

With this notation, we show that Branchlegro (i) is algebraically equivalent to dense gated evaluation, (ii) preserves energy gradients and Hessians, and therefore forces, and (iii) preserves the equivariance and permutation symmetries of Allegro.

B.1 Energy equivalence

For a layer equipped with a router, define the continuation gate

$$f^{ij,l} = \mathcal{F}(\nu^{ij,l}), \qquad \nu^{ij,l} = \mathrm{MLP}_{\mathrm{router}_l}\left(\mathbf{x}^{ij,l}\right), \tag{11}$$

with $f^{ij,l} \in [0,1]$. For layers without a router, we define $f^{ij,l} = 1$. The gated tensor feature is

$$\mathbf{V}^{ij,l} = f^{ij,l}\bar{\mathbf{V}}^{ij,l}, \tag{12}$$

and the gated TP is

$$\bar{\mathbf{V}}^{ij,l+1} = \mathbf{A}^{i,l} \otimes \mathbf{V}^{ij,l}, \qquad ij \in \mathcal{E}. \tag{13}$$

For energy equivalence, we require the following:

- **A1. Exact-zero slicing.** An edge is removed from TP computation only when its continuation gate is exactly zero.
- **A2. Dense scalar computation.** Scalar features $\mathbf{x}^{ij,l}$ and environment weights $w^{ij,l}$ remain evaluated for all edges. Only TP computation is sparsified.

Let $E_{\mathrm{dense}}(\mathbf{R})$ denote evaluation in which all edges remain in the TP computation at every layer, including edges with $f^{ij,l} = 0$. Thus,

$$\bar{\mathbf{V}}^{ij,l+1}_{\mathrm{dense}} = \mathbf{A}^{i,l}_{\mathrm{dense}} \otimes \mathbf{V}^{ij,l}_{\mathrm{dense}}, \qquad ij \in \mathcal{E}. \tag{14}$$

Let $E_{\mathrm{sparse}}(\mathbf{R})$ denote the corresponding sparse evaluation. Define the active edge sets recursively as

$$\mathcal{E}_0 = \mathcal{E}, \qquad \mathcal{E}_{l+1} = \left\{ij \in \mathcal{E}_l : f^{ij,l} > 0\right\}. \tag{15}$$

Sparse evaluation computes

$$\bar{\mathbf{V}}^{ij,l+1}_{\mathrm{sparse}} = \mathbf{A}^{i,l}_{\mathrm{sparse}} \otimes \mathbf{V}^{ij,l}_{\mathrm{sparse}}, \qquad ij \in \mathcal{E}_{l+1}, \tag{16}$$

while the TP is not evaluated for $ij \notin \mathcal{E}_{l+1}$.

Because subsequent scalar computation remains defined over the full edge set, the TP output is zero-padded as

$$\mathbf{z}^{ij,l+1}_{\mathrm{sparse}} = \begin{cases} \left(\bar{\mathbf{V}}^{ij,l+1}_{\mathrm{sparse}}\right)_{\ell=0}, & ij \in \mathcal{E}_{l+1}, \\ \mathbf{0}, & ij \notin \mathcal{E}_{l+1}. \end{cases} \tag{17}$$

For dense evaluation, $\mathbf{z}^{ij,l+1}_{\mathrm{dense}} = (\bar{\mathbf{V}}^{ij,l+1}_{\mathrm{dense}})_{\ell=0}$.

By Assumption **A2.**, Equation 9 is evaluated for all $ij \in \mathcal{E}$ in both the dense and sparse evaluations.

Proposition 1. *Under Assumptions **A1.** and **A2.**, dense gated evaluation and sparse active-set evaluation produce identical scalar latent features at every layer and therefore identical energies:*

$$E_{\mathrm{dense}}(\mathbf{R}) = E_{\mathrm{sparse}}(\mathbf{R}), \qquad \forall\, \mathbf{R}. \tag{18}$$

Proof. We proceed by induction over the layers. At layer l, assume that

$$\mathbf{x}^{ij,l}_{\mathrm{dense}} = \mathbf{x}^{ij,l}_{\mathrm{sparse}}, \qquad w^{ij,l}_{\mathrm{dense}} = w^{ij,l}_{\mathrm{sparse}}, \qquad \forall\, ij \in \mathcal{E}, \tag{19}$$

and that

$$\bar{\mathbf{V}}^{ij,l}_{\mathrm{dense}} = \bar{\mathbf{V}}^{ij,l}_{\mathrm{sparse}}, \qquad \forall\, ij \in \mathcal{E}_l, \tag{20}$$

while

$$\bar{\mathbf{V}}^{ij,l}_{\mathrm{dense}} = \mathbf{0}, \qquad \forall\, ij \notin \mathcal{E}_l. \tag{21}$$

These conditions hold initially because $\mathcal{E}_0 = \mathcal{E}$ and both evaluations begin from the same embedding.

Since the scalar features are identical, both evaluations produce the same continuation gates for all active edges. Moreover, by Assumption **A2.**, the environment weights $w^{ij,l}$ remain evaluated for every edge in both evaluations. Since these weights are identical by the induction hypothesis and the environment tensor is constructed over the full edge set,

$$\mathbf{A}^{i,l}_{\text{dense}} = \sum_{k:(i,k)\in\mathcal{E}} w^{ik,l}_{\text{dense}} \mathbf{Y}^{ik} = \sum_{k:(i,k)\in\mathcal{E}} w^{ik,l}_{\text{sparse}} \mathbf{Y}^{ik} = \mathbf{A}^{i,l}_{\text{sparse}}. \tag{22}$$

For an edge $ij \in \mathcal{E}_{l+1}$, both evaluations therefore apply the same gate to the same tensor feature and evaluate the same TP, giving by Equation 13

$$\bar{\mathbf{V}}^{ij,l+1}_{\text{dense}} = \bar{\mathbf{V}}^{ij,l+1}_{\text{sparse}}. \tag{23}$$

Now consider $ij \notin \mathcal{E}_{l+1}$. If the edge exits at layer l, then by Assumption **A1.**, $f^{ij,l} = 0$, and hence

$$\mathbf{V}^{ij,l}_{\text{dense}} = f^{ij,l} \bar{\mathbf{V}}^{ij,l}_{\text{dense}} = \mathbf{0}. \tag{24}$$

By bilinearity of the TP,

$$\bar{\mathbf{V}}^{ij,l+1}_{\text{dense}} = \mathbf{A}^{i,l}_{\text{dense}} \otimes \mathbf{0} = \mathbf{0}. \tag{25}$$

If the edge exited at an earlier layer, the induction hypothesis already gives $\bar{\mathbf{V}}^{ij,l}_{\text{dense}} = \mathbf{0}$, and the same conclusion follows. Thus, dense evaluation produces exactly zero for every TP output omitted by sparse evaluation. After zero-padding (Equation 17),

$$\mathbf{z}^{ij,l+1}_{\text{dense}} = \mathbf{z}^{ij,l+1}_{\text{sparse}}, \qquad \forall\, ij \in \mathcal{E}. \tag{26}$$

By Assumption **A2.**, the latent update in Equation 9 is evaluated for every edge in both implementations. Since its complete input is identical, it follows that

$$\mathbf{x}^{ij,l+1}_{\text{dense}} = \mathbf{x}^{ij,l+1}_{\text{sparse}}, \qquad w^{ij,l+1}_{\text{dense}} = w^{ij,l+1}_{\text{sparse}}, \qquad \forall\, ij \in \mathcal{E}. \tag{27}$$

This establishes the induction step.

Therefore the complete scalar latent histories are identical in dense and sparse evaluation. By the common energy readout defined in Equation 10,

$$E_{\text{dense}}(\mathbf{R}) = E_{\text{sparse}}(\mathbf{R}), \qquad \forall\, \mathbf{R}. \tag{28}$$

□

B.2 Derivative equivalence

To extend the equivalence to energy gradients, we additionally require:

A3. Smooth routing. The router probability map $\nu^{ij,l}$ and the continuation function $\mathcal{F}$ are C^2, and the zero set

$$\mathcal{Z} := \{u : \mathcal{F}(u) = 0\} \tag{29}$$

is an interval with nonempty interior.

The interior condition is essential: a continuation function vanishing only at an isolated point would satisfy $\mathcal{F}(\nu^\star) = 0$ but generally $\mathcal{F}'(\nu^\star) \neq 0$, leaving a nonzero derivative contribution in the omitted branch.

Proposition 2. *Under Assumptions **A1.–A3.**, differentiating the sparse Branchlegro computation yields the same energy gradients and Hessians as dense gated evaluation. Consequently,*

$$\mathbf{F}_{\text{sparse}}(\mathbf{R}) = -\nabla_{\mathbf{R}} E_{\text{sparse}}(\mathbf{R}) = -\nabla_{\mathbf{R}} E_{\text{dense}}(\mathbf{R}) = \mathbf{F}_{\text{dense}}(\mathbf{R}), \qquad \nabla^2_{\mathbf{R}} E_{\text{sparse}} = \nabla^2_{\mathbf{R}} E_{\text{dense}}. \tag{30}$$

Proof. By Proposition 1, under Assumptions **A1.** and **A2.**, the dense and sparse energy functions are pointwise identical. It therefore remains to show that differentiation through the sparse computation does not omit a nonzero derivative contribution when the active set changes.

Consider one gated tensor feature,

$$\mathbf{V}(\mathbf{R}) = f(\mathbf{R})\bar{\mathbf{V}}(\mathbf{R}) = \mathcal{F}(\nu(\mathbf{R}))\bar{\mathbf{V}}(\mathbf{R}). \tag{31}$$

By Assumption **A3.** and the twice continuous differentiability of the Allegro backbone, the gated feature is twice differentiable with respect to $\mathbf{R}$. By the chain rule,

$$\begin{aligned}
\nabla_{\mathbf{R}}\mathbf{V} &= \mathcal{F}'(\nu)\,(\nabla_{\mathbf{R}}\nu \otimes \bar{\mathbf{V}}) + \mathcal{F}(\nu)\,\nabla_{\mathbf{R}}\bar{\mathbf{V}}, \\
\nabla^2_{\mathbf{R}}\mathbf{V} &= \mathcal{F}''(\nu)\,(\nabla_{\mathbf{R}}\nu \otimes \nabla_{\mathbf{R}}\nu \otimes \bar{\mathbf{V}}) \\
&\quad + \mathcal{F}'(\nu)\left(\nabla^2_{\mathbf{R}}\nu \otimes \bar{\mathbf{V}} + 2\,\nabla_{\mathbf{R}}\nu \otimes \nabla_{\mathbf{R}}\bar{\mathbf{V}}\right) + \mathcal{F}(\nu)\,\nabla^2_{\mathbf{R}}\bar{\mathbf{V}}.
\end{aligned} \tag{32}$$

Let $\mathcal{Z}$ be the zero-valued slicing interval defined in Assumption **A3.**. We consider three cases.

If $\nu \notin \mathcal{Z}$, then $\mathcal{F}(\nu) \neq 0$ by definition of $\mathcal{Z}$. By Assumption **A1.**, no sparse slicing occurs, so the dense and sparse evaluations execute the same differentiable operations.

If $\nu \in \mathrm{int}(\mathcal{Z})$, then $\mathcal{F}$ is locally constant and $\mathcal{F}(\nu) = 0$, $\mathcal{F}'(\nu) = 0$, and $\mathcal{F}''(\nu) = 0$. Therefore, by Equation 32, $\nabla_{\mathbf{R}}\mathbf{V} = \mathbf{0}$ and $\nabla^2_{\mathbf{R}}\mathbf{V} = \mathbf{0}$. Thus, a branch omitted by sparse evaluation contributes neither to the forward value nor to its derivatives.

Finally, let $\nu^\star$ lie on the boundary of $\mathcal{Z}$. Since $\mathcal{F}$ is continuous, $\mathcal{Z}$ is closed and $\mathcal{F}(\nu^\star) = 0$. By Assumption **A3.**, $\mathcal{F}'$ and $\mathcal{F}''$ are continuous, and both vanish identically on $\mathrm{int}(\mathcal{Z})$, where $\mathcal{F}$ is constant. Therefore,

$$\mathcal{F}'(\nu^\star) = \lim_{\substack{u \to \nu^\star \\ u \in \mathrm{int}(\mathcal{Z})}} \mathcal{F}'(u) = 0, \qquad \mathcal{F}''(\nu^\star) = \lim_{\substack{u \to \nu^\star \\ u \in \mathrm{int}(\mathcal{Z})}} \mathcal{F}''(u) = 0. \tag{33}$$

Consequently, every term in Equation 32 vanishes, so $\nabla_{\mathbf{R}}\mathbf{V} = \mathbf{0}$ and $\nabla^2_{\mathbf{R}}\mathbf{V} = \mathbf{0}$. Hence, at the boundary where the sparse active set changes, the omitted branch has zero value, zero first derivative, and zero second derivative.

Thus, changing the sparse active set removes no nonzero first- or second-derivative contribution. All remaining downstream operations are shared between dense and sparse evaluation and are C^2 in Allegro, so by the chain rule their first and second derivatives with respect to $\mathbf{R}$ coincide. Together with Proposition 1, this gives

$$\nabla_{\mathbf{R}}E_{\mathrm{dense}}(\mathbf{R}) = \nabla_{\mathbf{R}}E_{\mathrm{sparse}}(\mathbf{R}), \qquad \nabla^2_{\mathbf{R}}E_{\mathrm{dense}}(\mathbf{R}) = \nabla^2_{\mathbf{R}}E_{\mathrm{sparse}}(\mathbf{R}), \tag{34}$$

and hence

$$\mathbf{F}_{\mathrm{dense}}(\mathbf{R}) = \mathbf{F}_{\mathrm{sparse}}(\mathbf{R}). \tag{35}$$

□

The same argument extends directly to stresses by replacing derivatives with respect to atomic coordinates by derivatives with respect to strain.

Corollary 1. *Under Assumptions **A1.–A3.**, the Branchlegro PES is C^2 in $\mathbf{R}$.*

Proof. By Assumption **A3.**, $\mathcal{F}$ and ν are C^2, and the Allegro backbone is C^2. The dense gated energy is therefore C^2 as a composition of C^2 maps. By Proposition 1, $E_{\mathrm{sparse}} = E_{\mathrm{dense}}$ pointwise, so the sparse energy inherits this regularity. □

The same argument applies at any order m for which $\mathcal{F}$, ν, and the backbone are C^m, and derivatives up to order m computed through the sparse path are exact.

B.3 Symmetry preservation

Proposition 3. *Given an $E(3)$-invariant continuation gate produced by a shared edge-wise router, Branchlegro preserves (i) the $E(3)$-equivariance of Allegro's tensor features and invariance of its energy, and (ii) its permutation symmetry.*

Proof. Let $g \in O(3)$ denote a rotation or reflection acting on the atomic configuration $\mathbf{R}$, and let $D[g]$ denote the corresponding representation acting on Allegro's equivariant tensor features. Translations act trivially since Allegro depends only on relative geometry. Let π denote a permutation of atom indices, acting on edge labels as $ij \mapsto \pi(i)\pi(j)$.

$E(3)$-equivariance. We proceed inductively over layers. Initially, $\mathcal{E}_0(\mathbf{R}) = \mathcal{E} = \mathcal{E}_0(g\mathbf{R})$. Assume at layer l that the active sets coincide and that $\bar{\mathbf{V}}^{ij,l}$ is equivariant. By invariance of the continuation gate, $f^{ij,l}(g\mathbf{R}) = f^{ij,l}(\mathbf{R})$. Thus,

$$\begin{aligned}\mathbf{V}^{ij,l}(g\mathbf{R}) &= f^{ij,l}(g\mathbf{R})\bar{\mathbf{V}}^{ij,l}(g\mathbf{R}) \\ &= f^{ij,l}(\mathbf{R})D[g]\bar{\mathbf{V}}^{ij,l}(\mathbf{R}) \\ &= D[g]\mathbf{V}^{ij,l}(\mathbf{R}).\end{aligned} \tag{36}$$

Moreover,

$$\begin{aligned}\mathcal{E}_{l+1}(g\mathbf{R}) &= \{ij \in \mathcal{E}_l(g\mathbf{R}) : f^{ij,l}(g\mathbf{R}) > 0\} \\ &= \{ij \in \mathcal{E}_l(\mathbf{R}) : f^{ij,l}(\mathbf{R}) > 0\} = \mathcal{E}_{l+1}(\mathbf{R}).\end{aligned} \tag{37}$$

The subsequent Allegro operations are equivariant, so $\bar{\mathbf{V}}^{ij,l+1}$ retains the corresponding transformation law. The result therefore follows by induction, with the scalar energy readout remaining $E(3)$-invariant.

Permutation symmetry. We proceed inductively over layers. Initially, $\mathcal{E}_0(\pi\mathbf{R}) = \pi(\mathcal{E}_0(\mathbf{R}))$. Assume at layer l that the active sets and incoming tensor features permute consistently,

$$\mathcal{E}_l(\pi\mathbf{R}) = \pi(\mathcal{E}_l(\mathbf{R}))\,, \qquad \bar{\mathbf{V}}^{\pi(i)\pi(j),l}(\pi\mathbf{R}) = \bar{\mathbf{V}}^{ij,l}(\mathbf{R}).$$

Since Allegro's edge features permute consistently and the same router is applied to every edge, $f^{\pi(i)\pi(j),l}(\pi\mathbf{R}) = f^{ij,l}(\mathbf{R})$. Thus,

$$\begin{aligned}\mathbf{V}^{\pi(i)\pi(j),l}(\pi\mathbf{R}) &= f^{\pi(i)\pi(j),l}(\pi\mathbf{R})\bar{\mathbf{V}}^{\pi(i)\pi(j),l}(\pi\mathbf{R}) \\ &= f^{ij,l}(\mathbf{R})\bar{\mathbf{V}}^{ij,l}(\mathbf{R}) \\ &= \mathbf{V}^{ij,l}(\mathbf{R}).\end{aligned} \tag{38}$$

Moreover,

$$\begin{aligned}\mathcal{E}_{l+1}(\pi\mathbf{R}) &= \{\pi(i)\pi(j) \in \mathcal{E}_l(\pi\mathbf{R}) : f^{\pi(i)\pi(j),l}(\pi\mathbf{R}) > 0\} \\ &= \pi\big(\{ij \in \mathcal{E}_l(\mathbf{R}) : f^{ij,l}(\mathbf{R}) > 0\}\big) \\ &= \pi(\mathcal{E}_{l+1}(\mathbf{R}))\,.\end{aligned} \tag{39}$$

The subsequent Allegro operations preserve permutation symmetry, so $\bar{\mathbf{V}}^{ij,l+1}$ permutes consistently as well. The result therefore follows by induction, and the final sum over edge energies is permutation invariant. □

In Branchlegro, each router is an MLP acting only on Allegro's invariant scalar edge features $\mathbf{x}^{ij,l}$. Therefore its output $\nu^{ij,l}$, and consequently the continuation gate $f^{ij,l} = \mathcal{F}(\nu^{ij,l})$, are $E(3)$-invariant.

Together, Propositions 1–3 and Corollary 1 establish Theorem 1: Assumptions **A1.** and **A2.** state condition (ii) for Allegro's edge-wise routing, Assumption **A3.** states the smoothness in (i), and the invariance in (i) follows from the routers acting on invariant scalar features.

C Training Setup

All model training was conducted on a combination of NVIDIA A100-SXM4-40GB, NVIDIA A100-SXM4-80GB, and H200 GPUs. For fine-tuning the FMs, the models were compiled and employed cuEquivariance TP kernels. The hyperparameters of the bespoke Allegro models for CO/Pt and CDP are summarized in Table 2a and Table 4a. The corresponding hyperparameters of the Allegro and Branchlegro training are reported in Tables 2b–2c for CO/Pt and Tables 4b–4c for CDP. Across Branchlegro runs, the random seed controls router initialization and data-batch ordering; the teacher model and train/validation/test split were held fixed.

Table 2: Bespoke model architecture and training hyperparameters used for CO/Pt.

(a) Architecture (Owen et al., 2024).

HYPERPARAMETER	VALUE
CUTOFF RADIUS	8.0 Å
NUMBER OF LAYERS	1 / 2
$l_{\max}$	4
PARITY	TRUE
NUMBER OF SCALAR FEATURES	256
NUMBER OF TENSOR FEATURES	16
RADIAL BESSEL BASIS FUNCTIONS	8
TRAINABLE BESSEL BASIS	FALSE
POLYNOMIAL CUTOFF EXPONENT	6
RADIAL-CHEMICAL EMBEDDING DIM	64
SCALAR EMBEDDING MLP DEPTH	1
SCALAR EMBEDDING MLP WIDTH	128
SCALAR EMBEDDING NONLINEARITY	SILU
ALLEGRO MLP DEPTH	3
ALLEGRO MLP WIDTH	256
ALLEGRO MLP NONLINEARITY	SILU
TP PATH-CHANNEL COUPLING	FALSE
READOUT MLP DEPTH	1
READOUT MLP WIDTH	256
READOUT MLP NONLINEARITY	SILU
PER-TYPE ENERGY SCALES TRAINABLE	FALSE
PER-TYPE ENERGY SHIFTS TRAINABLE	FALSE

(b) Allegro training.

HYPERPARAMETER	VALUE
EPOCHS	1000
OPTIMIZER	ADAM
LEARNING RATE	2×10^{-3}
LR SCHEDULER	REDUCE ON PLATEAU
EARLY STOPPING	PATIENCE 300
FORCE LOSS COEFF.	1
ENERGY LOSS COEFF.	1
STRESS LOSS COEFF.	1
BATCH SIZE	2

(c) Branchlegro distillation.

HYPERPARAMETER	VALUE
EPOCHS	100
OPTIMIZER	ADAM
BACKBONE LEARNING RATE	1×10^{-3}
ROUTER LEARNING RATE	5×10^{-4}
LR SCHEDULER	COSINE ANNEALING
FORCE LOSS COEFF.	1
ENERGY LOSS COEFF.	1
STRESS LOSS COEFF.	1
$\mathcal{L}_{\text{distill}}$ COEFF.	0.05
CONTINUE TOLERANCE	$0 : 0.060,\ 1 : 0.095$
STOP TOLERANCE	$0 : 0.12,\ 1 : 0.155$
ROUTER MLP DEPTH	2
ROUTER MLP WIDTH	8
BATCH SIZE	8

Table 3: FM fine-tuning hyperparameters used for CO/Pt (architecture from Kavanagh et al. (2026)).

(a) Allegro-OAM-L fine-tuning.

HYPERPARAMETER	VALUE
EPOCHS	1000
OPTIMIZER	ADAMW
LEARNING RATE	4.1×10^{-4}
WEIGHT DECAY	3.4×10^{-7}
LR SCHEDULER	COSINE ANNEALING
EARLY STOPPING	PATIENCE 600
FORCE LOSS COEFF.	1
ENERGY LOSS COEFF.	1
STRESS LOSS COEFF.	1
BATCH SIZE	5

(b) Branchlegro distillation.

HYPERPARAMETER	VALUE
EPOCHS	200
OPTIMIZER	ADAM
BACKBONE LEARNING RATE	1×10^{-3}
ROUTER LEARNING RATE	1×10^{-4}
LR SCHEDULER	COSINE ANNEALING
FORCE LOSS COEFF.	1
ENERGY LOSS COEFF.	1
STRESS LOSS COEFF.	1
$\mathcal{L}_{\text{distill}}$ COEFF.	0.1
CONTINUE TOLERANCE	$0 : 0.030,\ 1 : 0.042,$ $2 : 0.048,\ 3 : 0.060$
STOP TOLERANCE	$0 : 0.060,\ 1 : 0.071,$ $2 : 0.077,\ 3 : 0.090$
ROUTER MLP DEPTH	1
ROUTER MLP WIDTH	8
BATCH SIZE	10

Table 4: Bespoke model architecture and training hyperparameters used for CDP.

(a) Architecture (Wang et al., 2025).

HYPERPARAMETER	VALUE
CUTOFF RADIUS	7.0 Å
NUMBER OF LAYERS	1 / 2
$l_{\max}$	2
PARITY	TRUE
NUMBER OF SCALAR FEATURES	32
NUMBER OF TENSOR FEATURES	32
RADIAL BESSEL BASIS FUNCTIONS	8
TRAINABLE BESSEL BASIS	FALSE
POLYNOMIAL CUTOFF EXPONENT	6
RADIAL-CHEMICAL EMBEDDING DIM	32
SCALAR EMBEDDING MLP DEPTH	2
SCALAR EMBEDDING MLP WIDTH	128
SCALAR EMBEDDING NONLINEARITY	SILU
ALLEGRO MLP DEPTH	2
ALLEGRO MLP WIDTH	128
ALLEGRO MLP NONLINEARITY	SILU
TP PATH-CHANNEL COUPLING	FALSE
READOUT MLP DEPTH	1
READOUT MLP WIDTH	32
READOUT MLP NONLINEARITY	NONE
PER-TYPE ENERGY SCALES TRAINABLE	TRUE
PER-TYPE ENERGY SHIFTS TRAINABLE	TRUE

(b) Allegro training (Harari et al., 2026).

HYPERPARAMETER	VALUE
EPOCHS	1000
OPTIMIZER	ADAMW
LEARNING RATE	3×10^{-3}
WEIGHT DECAY	1×10^{-4}
LR SCHEDULER	COSINE ANNEALING
EARLY STOPPING	PATIENCE 600
FORCE LOSS COEFF.	10
ENERGY LOSS COEFF.	1
BATCH SIZE	5

(c) Branchlegro distillation.

HYPERPARAMETER	VALUE
EPOCHS	500
OPTIMIZER	ADAM
BACKBONE LEARNING RATE	5×10^{-3}
ROUTER LEARNING RATE	5×10^{-4}
LR SCHEDULER	COSINE ANNEALING
FORCE LOSS COEFF.	1
ENERGY LOSS COEFF.	1
$\mathcal{L}_{\text{distill}}$ COEFF.	0.1
CONTINUE TOLERANCE	$0 : 0.05,\ 1 : 0.08$
STOP TOLERANCE	$0 : 0.08,\ 1 : 0.12$
ROUTER MLP DEPTH	1
ROUTER MLP WIDTH	8
BATCH SIZE	5

Table 5: FM fine-tuning hyperparameters used for CDP (architecture from Kavanagh et al. (2026)).

(a) Allegro-OAM-L fine-tuning.

HYPERPARAMETER	VALUE
EPOCHS	1000
OPTIMIZER	ADAMW
LEARNING RATE	1.0×10^{-3}
WEIGHT DECAY	2.3×10^{-7}
LR SCHEDULER	COSINE ANNEALING
EARLY STOPPING	PATIENCE 600
FORCE LOSS COEFF.	10
ENERGY LOSS COEFF.	1
BATCH SIZE	5

(b) Branchlegro distillation.

HYPERPARAMETER	VALUE
EPOCHS	500
OPTIMIZER	ADAM
BACKBONE LEARNING RATE	5×10^{-4}
ROUTER LEARNING RATE	5×10^{-5}
LR SCHEDULER	COSINE ANNEALING
FORCE LOSS COEFF.	1
ENERGY LOSS COEFF.	1
$\mathcal{L}_{\text{distill}}$ COEFF.	0.05
CONTINUE TOLERANCE	$0 : 0.05,\ 1 : 0.07,$ $2 : 0.09,\ 3 : 0.11$
STOP TOLERANCE	$0 : 0.10,\ 1 : 0.12,$ $2 : 0.14,\ 3 : 0.16$
ROUTER MLP DEPTH	1
ROUTER MLP WIDTH	8
BATCH SIZE	4

D MD THROUGHPUT AND MEMORY MEASUREMENTS

Molecular dynamics throughput was benchmarked in LAMMPS (Thompson et al., 2022; Johansson et al., 2025) using NVT simulations at 525 K with a 0.5 fs timestep for CDP and 300 K with a 1 fs timestep for CO/Pt. Each benchmark used an initial 10^4-step warm-up period followed by a production run of up to 9.99×10^6 steps. LAMMPS performance statistics were recorded every 100 steps, and the benchmark run was limited to 1200 s to keep comparisons consistent across models and system sizes. Throughput was evaluated from the reported simulation performance after the warm-up period. Pareto benchmarks were run on NVIDIA A100-SXM4-40GB; all remaining benchmarks were run on NVIDIA A100-SXM4-80GB GPUs.

GPU memory was measured using PyTorch's CUDA memory profiler. Before each force evaluation in LAMMPS, the CUDA peak memory counters were reset. After the force evaluation, the peak allocated GPU memory were recorded. Reported values correspond to the maximum over all force evaluations during the simulation.

E PHYSICAL FIDELITY

E.1 ENERGY PROFILE FOR CO/PT

Figure 7 includes energy profiles of the Allegro and Branchlegro CO/Pt MLIPs against the density functional theory (DFT) ground truth. This includes energy-volume equation of state of bulk Pt and an isolated-CO bond-stretch curve. The DFT calculations followed the methodology reported by Owen et al. (2024), using VASP with PAW potentials, the vdW-DF exchange-correlation functional, and a 500 eV plane-wave cutoff. The Pt validation supercell used a 3×3×3 supercell of a conventional FCC cell with an optimized Pt lattice constant of approximately 4.03 Å. We used metallic sampling and smearing for Pt, molecular sampling and dipole corrections for CO, and consistent electronic convergence settings.

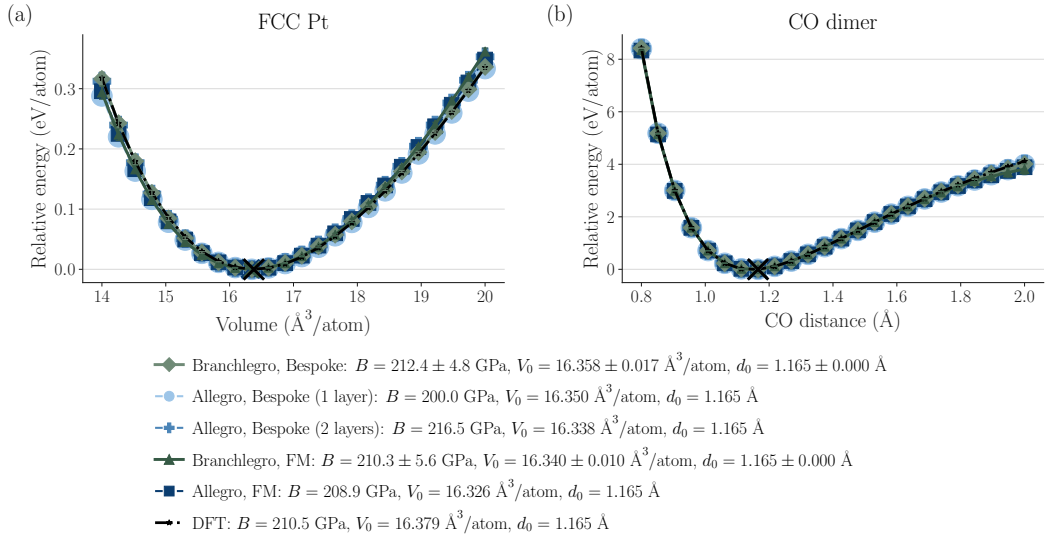


Figure 7: (a) Equation of state of fcc Pt, energy per atom as a function of atomic volume. Equilibrium volumes (V_0) and bulk moduli (B) were obtained using ASE's stabilized-jellium equation-of-state fit. (b) Absolute energy of an isolated CO dimer as a function of bond length, with d_0 denoting the lowest-energy sampled distance. Crosses mark the corresponding energy minima. Branchlegro results are reported as mean ± standard deviation over seven seeds.

Overall, all models show close agreement with DFT across the physical-fidelity evaluations. Both Allegro and Branchlegro reproduce the relevant energy landscapes and derived material properties, indicating that adaptive routing preserves the underlying physical response of the parent models.

Using the DFT bulk modulus, $B_{\mathrm{DFT}} = 210.5$ GPa, as reference, the bespoke one- and two-layer Allegro models deviate by -4.97% and $+2.85\%$. Branchlegro predicts $B = 212.4$ GPa, corresponding to a smaller deviation of $+0.90\%$ than either bespoke baseline. For the FMs, both Allegro and Branchlegro remain within 1% of DFT, with deviations of -0.75% and -0.10%, respectively. Thus, adaptive routing overall preserves the elastic response of both bespoke and FMs.

E.2 MD SIMULATIONS FOR CDP ACTIVATION ENERGY

To further validate Branchlegro dynamics for CDP, molecular dynamics (MD) simulations were performed in LAMMPS. We compare radial distribution functions (RDFs) from Branchlegro and Allegro models with *ab initio* MD (AIMD) using the DFT method used to generate the CDP training dataset Wang et al. (2025). The AIMD RDF reference averages two 144-atom trajectories that differ only in their initial velocity seed. For the bespoke models, Branchlegro, L1, and L2 Allegro models, RDFs are computed from the average of three 525 K MD simulations that differ only in their initial velocity seed with 1000 atoms. For the more computationally expensive FM models the RDFs are computed from the average of three 144-atom trajectories at 525 K that differ only in their initial velocity seed.

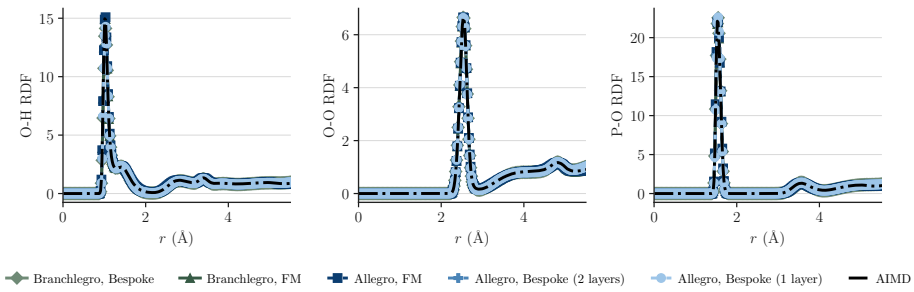


Figure 8: Radial distribution functions for O-H, O-O, and P-O pairs in molecular dynamics (MD) simulations of CDP at 525 K with *ab initio* MD reference.

Along with RDFs, we calculate the proton diffusion activation energy, a validation requiring accurate long-timescale MD dynamics. Simulations were performed in LAMMPS using the L1 Allegro, L2 Allegro, and Branchlegro bespoke interatomic potentials. Simulations were conducted in the NVT ensemble with periodic boundary conditions in all three dimensions with 1000 atoms and a Nosé–Hoover thermostat. Initial velocities were drawn from a Gaussian distribution at the target temperature. A timestep of 0.5 fs was used, and each simulation was run for 4×10^6 steps, corresponding to 2 ns of simulated time. Atomic configurations were recorded every 100 steps.

Hydrogen diffusion was determined from the mean-squared displacement (MSD) of H atoms averaged over three trajectories at each temperature that differ only in the initial velocity seed. The diffusion coefficient was obtained from the linear diffusive regime using the three-dimensional Einstein relation. The initial 10 ps of each trajectory were excluded from the MSD analysis. Diffusion coefficients were calculated at 525 K, 550 K, 575 K, and 625 K for each potential. Activation energies were obtained by fitting the resulting temperature-dependent diffusion coefficients to the Arrhenius relation. We restrict activation energy calculations to the bespoke models because the required long-timescale simulations are extremely expensive for the FMs.

F ENERGY CONSERVATION IN MD

For a correct MD simulation, the total energy (summation of potential and kinetic energies) should remain constant over time in the microcanonical (NVE) ensemble. To evaluate the energy conservation of Branchlegro, we run NVE MD simulations in LAMMPS for all Branchlegro models. Simulations were initialized at 525 K and 300 K with periodic boundary conditions for CDP and CO/Pt and a timestep of 0.5 fs and 1 fs (production timesteps), and were propagated for 100 ps. The total energy was recorded every 100 steps and converted to an energy drift per atom relative to the initial value,

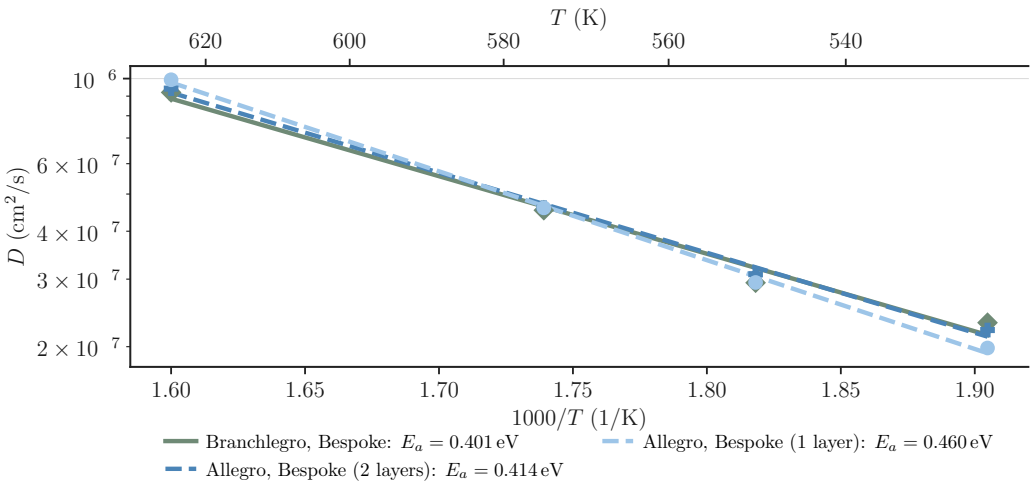


Figure 9: Arrhenius plot of diffusion coefficients for L1 Allegro, L2 Allegro, and Branchlegro bespoke models. The activation energy, reported in the legend, is determined from the slope of the fitted linear regression.

after an initial 10 ps equilibration period. Energy conservation was assessed by monitoring this change in total energy as a function of simulation time for the different Branchlegro and Allegro models.

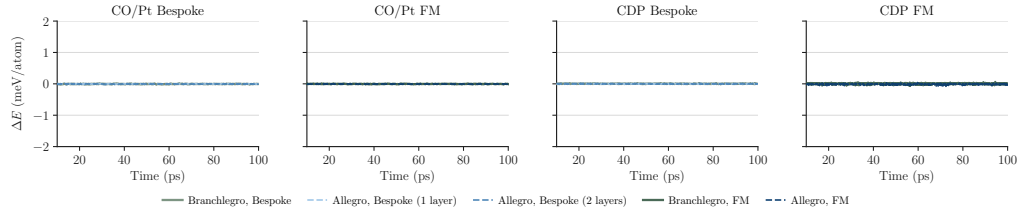


Figure 10: Energy conservation during molecular dynamics in the microcanonical ensemble. Each panel shows the drift in total energy, reported as the change in total energy in meV per atom relative to the value at 10 ps. All cases show stable and conservative dynamics.

G Interpretability Details

The CO/Pt analysis utilized the following classification rules: A Pt is adsorbate-bound if a C atom lies within 2.6 Å[1]; otherwise it is bare surface if its Pt–Pt coordination within 3.2Å (tail of first peak in Pt–Pt RDF, Figure 5c) is below the fcc bulk value of 12, and bulk if not. Adsorbate-bound Pt is further split by the coordination of its adsorbed carbon under the same 2.6 Å criterion: top if that C binds a single Pt, multiple if it is shared with more than one Pt (bridge or hollow). Figure 11 shows the different adsorption sites for CO in Pt.

The CDP analysis utilized the following classification rules: For the P–H edges, covalent was classified as P owning the H's closest O atom. Hydrogen-bonded was classified when that P owns the H's second nearest O and that O is within 2.20 Å (the minimum after the second peak in the O–H RDF). Otherwise, it was classified as unrelated. For O–O edges, same PO_4 was classified when both O atoms share the same closest P atom. O-pair was classified when the two O atoms are the closest and second-nearest oxygens of a common proton, the latter again within 2.20 Å, so that a proton bridges them (covalently bonded to one and hydrogen-bonded to the other).

[1]Determined by JmolNN from Pan et al. (2021) with the radii taken from Cordero et al. (2008).

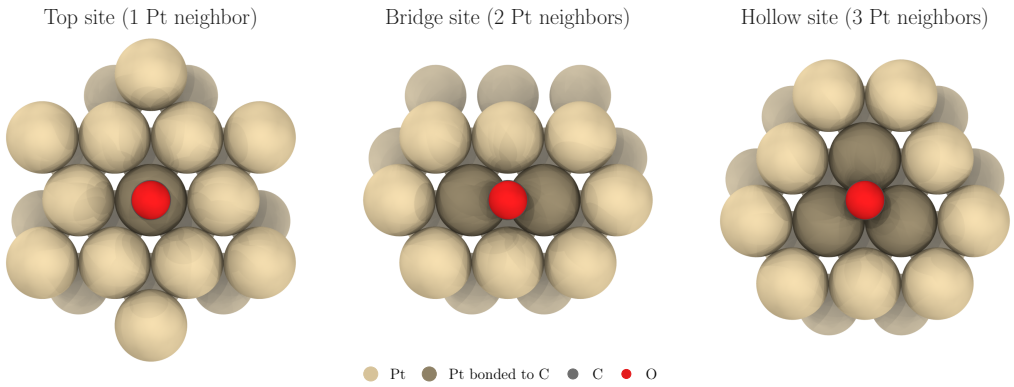


Figure 11: Schematic of CO adsorption sites on Pt.

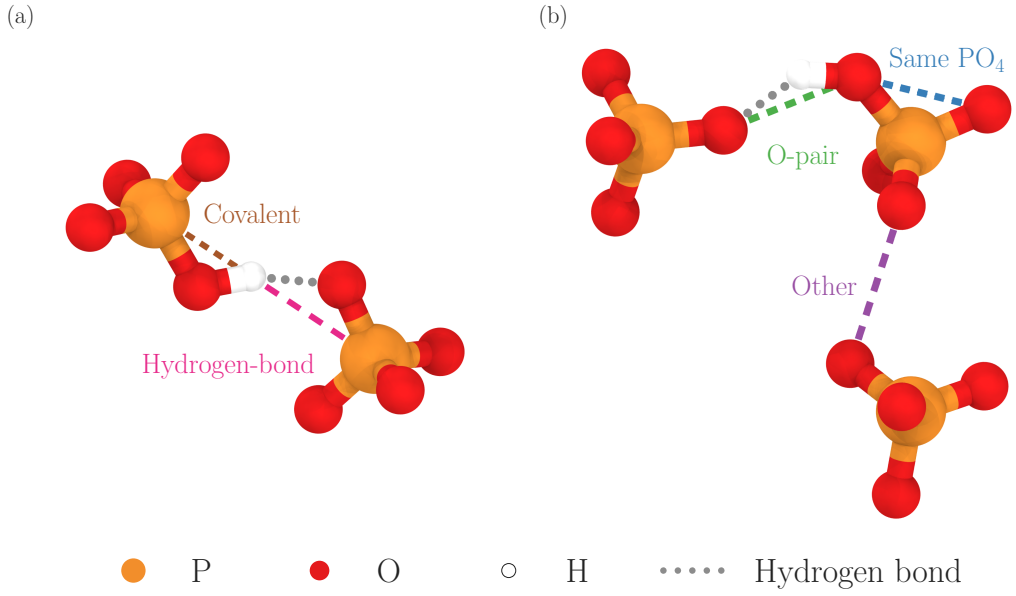


Figure 12: Edge classifications in CDP.