RETHINKING EDGE-WISE COMPUTATION FOR EFFICIENT EQUIVARIANT INTERATOMIC POTENTIALS

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern equivariant machine learning interatomic potentials are becoming increasingly computationally demanding as their architectures grow more sophisticated. In these architectures, edge-wise operations often account for a substantial portion of the overall computational cost. However, increasing edge-wise complexity is not necessarily the most efficient way to improve model accuracy. Keeping edge-wise computation simple enables effective use of modern fused kernels, substantially improving computational throughput. We therefore introduce EquFlash-S, an equivariant architecture that combines lightweight edge-wise message passing with expressive nonlinear processing in a separate residual node-wise feed-forward block. Its low per-edge cost enables scaling to larger model capacities for higher accuracy while retaining substantially higher throughput than more complex edge-wise designs. EquFlash-S achieves competitive performance across diverse atomistic benchmarks, including Matbench Discovery, the MDR Phonon Benchmark, and OC20. In particular, on Matbench Discovery, EquFlash-S achieves a Combined Performance Score of 0.907 while delivering 2.7–5.7$\times$ higher throughput than state-of-the-art models with comparable accuracy.

1 INTRODUCTION

Machine learning interatomic potentials (MLIPs) have been developed to accelerate first-principles calculations, enabling simulations that overcome the spatial and temporal limitations of conventional quantum-mechanical methods while approaching first-principles accuracy. Early GNN-based approaches enriched rotation-invariant representations with higher-order geometric information, ranging from interatomic distances to bond angles and dihedral angles (Schütt et al., 2017; Gasteiger et al., 2020; 2021). Meanwhile, equivariant GNNs employed Clebsch–Gordan tensor products (CGTPs) to propagate features with higher rotation orders (Batzner et al., 2022; Batatia et al., 2022).

Such equivariant GNNs have achieved considerable success in developing accurate interatomic potentials (Park et al., 2024; Batatia et al., 2025). Despite these advances, the computational cost of conventional CGTPs remains a significant challenge, particularly as the maximum rotation order $\ell_{\max}$ increases. This has motivated algorithmic reformulations of tensor products. In particular, eSCN (Passaro & Zitnick, 2023) introduced an edge-aligned local frame that reduces the computational complexity of the tensor product from $O({\ell_{\max}}^6)$ to $O({\ell_{\max}}^3)$.

These advances in tensor product efficiency have enabled equivariant architectures to perform increasingly expressive processing on individual edges, combining multiple equivariant transformations, nonlinearities, or attention operations before neighbor aggregation (Liao et al., 2024; 2026; Fu et al., 2025; Wood et al., 2026). Similar trends can also be found in rotation-invariant (Gasteiger et al., 2021; Zhou et al., 2026) and unconstrained architectures (Pozdnyakov & Ceriotti, 2023; Neumann et al., 2024; Rhodes et al., 2025), where multiple feature transformations are performed on edges. While such designs provide strong expressivity, their computational and memory costs scale with the number of edges. As the number of edges can far exceed the number of nodes, computational costs can become prohibitive in large-scale MLIP simulations. This brings us back to a basic architectural question: how much expressivity needs to be placed in edge-wise message computation? Specifically, rather than increasing the complexity of the edge-wise message function, can simple equivariant message passing retain sufficient predictive power?

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.

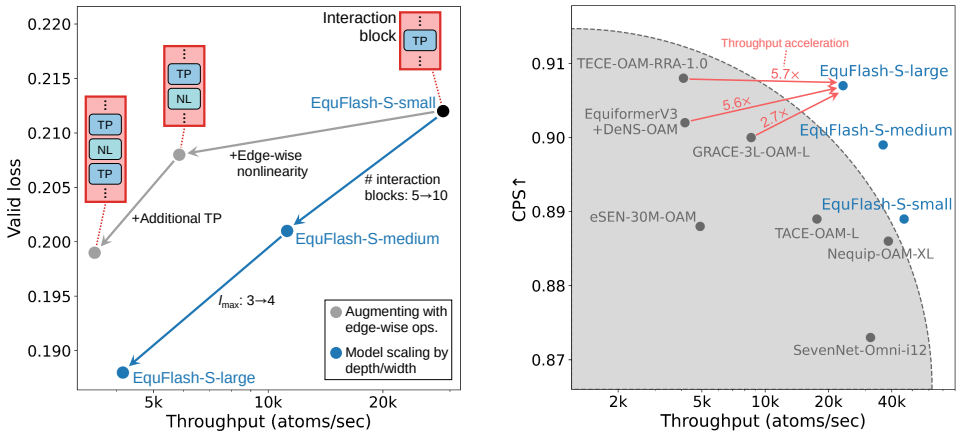

Figure 1: Accuracy–efficiency trade-offs in equivariant MLIPs. The left panel shows validation loss versus training throughput as edge-wise convolution expressivity increases or model capacity is scaled in depth and $\ell_{\max}$. TP and NL denote tensor product and nonlinearity, respectively. The right panel shows CPS versus peak inference throughput on Matbench Discovery, comparing EquFlash-S with state-of-the-art MLIPs.

In this work, we revisit CGTP-based architectures and report two key findings. First, we find that scaling CGTP-based designs, when paired with modern fused equivariant kernels (NVIDIA, 2025; Lee et al., 2025; Bharadwaj et al., 2025), can achieve a more favorable accuracy–efficiency trade-off than increasing edge-wise expressivity. As illustrated in the left panel of Fig. 1, scaling simple CGTP-based architectures in depth and $\ell_{\max}$ achieves lower validation loss at higher throughput than introducing additional edge-wise nonlinearities and tensor products in the interaction blocks. This advantage arises from the lightweight message-passing structure, which allows fused kernels to jointly perform tensor product and neighbor aggregation without materializing intermediate high-order edge features. Second, we find that the design of interaction blocks benefits from combining a residual message-passing sub-block with a separate residual nonlinear node-wise feed-forward sub-block. With this decoupled design, predictive accuracy consistently improves as model capacity increases, despite its simplicity.

Based on these findings, we introduce **EquFlash-S**, a simple yet effective equivariant architecture that combines lightweight CGTP-based message passing with residual node-wise nonlinear processing, rather than introducing additional edge-wise operations. As shown in the right panel of Fig. 1, the proposed model achieves a near-state-of-the-art Combined Performance Score (CPS) of 0.907 on Matbench Discovery (Riebesell et al., 2025) while delivering 2.7–5.7× higher throughput than other equivariant models with a CPS of 0.900 or higher.

2 Background

Given an atomic configuration of n atoms $\mathcal{X} = (\mathbf{Z}, \mathbf{r}, \mathbf{C})$, where $\mathbf{Z} = (Z_1, \ldots, Z_n) \in \mathbb{N}^n$ denotes the atomic numbers, $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_n) \in \mathbb{R}^{n\times 3}$ denotes the atomic positions, and $\mathbf{C} \in \mathbb{R}^{3\times 3}$ contains the lattice vectors for periodic systems, an MLIP predicts the total energy $E \in \mathbb{R}$, atomic forces $\mathbf{f} = (\mathbf{f}_1, \ldots, \mathbf{f}_n) \in \mathbb{R}^{n\times 3}$, and, for periodic systems, the stress tensor $\boldsymbol{\sigma} \in \mathbb{R}^{3\times 3}$.

The Euclidean group in three dimensions, denoted by $E(3)$, consists of translations, rotations, and reflections. The predictions of an MLIP should transform consistently under these operations: the energy remains invariant, while forces and stress transform equivariantly. Accordingly, we consider $E(3)$-equivariant neural networks, in which geometric features are represented by irreducible representations indexed by a non-negative integer rotation order ℓ and parity $p \in \{e, o\}$. We restrict the feature space to irreducible representations with natural parity, i.e., $0e, 1o, 2e, 3o, \ldots$, and allow multiple channels for each representation. Translation invariance is ensured by using relative atomic positions, while inversion acts on each natural-parity representation of order ℓ by a factor of $(-1)^\ell$. We therefore describe the transformation of features explicitly only under proper rotations in SO(3). For notational simplicity, we omit both the parity and channel indices hereafter.

An irreducible representation of rotation order ℓ is a $(2\ell+1)$-dimensional vector, $\mathbf{x}^{(\ell)} \in \mathbb{R}^{2\ell+1}$, which transforms under a rotation $\mathbf{R} \in \mathrm{SO}(3)$ according to $\mathbf{x}^{(\ell)} \mapsto \mathbf{D}^{(\ell)}(\mathbf{R})\mathbf{x}^{(\ell)}$, where $\mathbf{D}^{(\ell)}(\mathbf{R}) \in \mathbb{R}^{(2\ell+1)\times(2\ell+1)}$ is the Wigner D-matrix of rotation order ℓ.

CGTP is employed as a basic equivariant operation that combines two irreducible representations $\mathbf{x}^{(\ell_1)} \in \mathbb{R}^{2\ell_1+1}$ and $\mathbf{y}^{(\ell_2)} \in \mathbb{R}^{2\ell_2+1}$, of rotation orders ℓ_1 and ℓ_2, respectively, into an output representation $\mathbf{z}^{(\ell_{\text{out}})} \in \mathbb{R}^{2\ell_{\text{out}}+1}$ of rotation order ℓ_{out} satisfying $|\ell_1 - \ell_2| \le \ell_{\text{out}} \le \ell_1 + \ell_2$. For each admissible output order, the tensor product is defined as

$$\mathbf{z}_m^{(\ell_{\text{out}})} = \sum_{m_1=-\ell_1}^{\ell_1} \sum_{m_2=-\ell_2}^{\ell_2} C_{\ell_1,m_1,\ell_2,m_2}^{\ell_{\text{out}},m}\, \mathbf{x}_{m_1}^{(\ell_1)}\, \mathbf{y}_{m_2}^{(\ell_2)}, \qquad m = -\ell_{\text{out}}, \ldots, \ell_{\text{out}},$$

where $C_{\ell_1,m_1,\ell_2,m_2}^{\ell_{\text{out}},m}$ denotes the Clebsch–Gordan coefficient. For notational simplicity, we write $\mathbf{X} \otimes^{SO(3)} \mathbf{Y}$ to denote the CGTP between $\mathbf{X}$ and $\mathbf{Y}$, restricted to tensor product paths that produce output irreducible representations with $\ell_{\text{out}} \le \ell_{\max}$ and natural parity.

3 Related Work

We focus here on equivariant architectures and their computational efficiency, while providing a broader discussion of related work, including MLIP architectures, datasets, and benchmarks, in Appendix A.2. **Equivariant Tensor Product Networks:** Rotation-equivariant neural networks have been developed through various approaches (Thomas et al., 2018; Anderson et al., 2019; Weiler et al., 2018; Fuchs et al., 2020). Building on Tensor Field Networks (Thomas et al., 2018), NequIP (Batzner et al., 2022) applied CGTPs to equivariant message passing for interatomic potentials. MACE (Batatia et al., 2022) employed symmetric contractions to construct higher-body-order features after neighbor aggregation. Allegro (Musaelian et al., 2023) developed a strictly local equivariant architecture for scalable molecular dynamics, while SevenNet (Park et al., 2024) extended NequIP-based potentials to efficient parallel simulations. The improved expressivity of CGTP networks, however, comes with considerable computational and memory costs. CGTPs couple multiple input and output irreducible representations, and their cost increases rapidly with channel width and the maximum representation degree. **Algorithmic Reformulation of Equivariant Operations:** To reduce the computational cost of tensor products, eSCN (Passaro & Zitnick, 2023) reformulated SO(3)-equivariant convolutions as efficient edge-aligned SO(2) convolutions. This formulation was subsequently adopted by eSEN (Fu et al., 2025) and UMA (Wood et al., 2026) with nonlinear edge-wise processing, while EquiformerV2 (Liao et al., 2024) and EquiformerV3 (Liao et al., 2026) combined it with equivariant attention. More recently, TECE (Xu et al., 2026) introduced higher-order edge-wise interactions and radial rotary attention. These developments motivate us to revisit whether increasingly expressive edge-wise processing is necessary for competitive predictive accuracy. **Kernel Acceleration for Tensor Products:** cuEquivariance (NVIDIA, 2025), OpenEquivariance (Bharadwaj et al., 2025), and FlashTP (Lee et al., 2025) accelerate CGTPs by exploiting sparsity and kernel fusion, including fused tensor-product evaluation and neighbor aggregation.

4 Methods

4.1 Architecture Overview

EquFlash-S (Figure 2) consists of an input embedding layer, a stack of interaction blocks, and a global pooling layer. Given an atomic configuration, it constructs a local neighborhood graph by connecting pairs of atoms within a cutoff radius. Node features are initialized from atomic species, while radial and angular edge features are constructed from interatomic distances and directions, respectively. Each interaction block consists of equivariant message passing followed by nonlinear node-wise feature transformation. The global pooling layer, with the preceding linear operations, projects the final node representations onto invariant scalar features, which

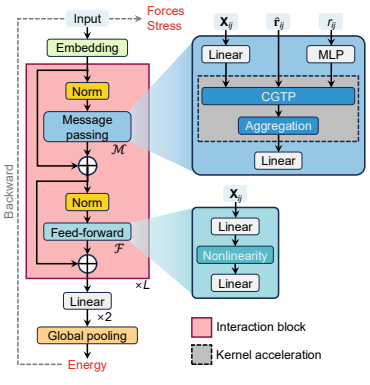

Figure 2: EquFlash-S architecture.

are summed to obtain the total energy. Unless otherwise specified, atomic forces are computed as the negative gradients of the predicted total energy with respect to atomic positions, yielding a conservative force field. For periodic systems, the stress tensor is obtained from the derivative of the total energy with respect to strain. Further architectural details are provided in Appendix A.1.

4.2 Interaction Block

Each interaction block consists of a message-passing layer followed by a node-wise feed-forward layer, both with residual connections. Let $\mathbf{X}^n$ denote the collection of node features entering the n-th interaction block, and let $\mathbf{r}_{ji}$ denote the relative displacement vector from atom j to atom i, with $r_{ji} = |\mathbf{r}_{ji}|$ and $\hat{\mathbf{r}}_{ji} = \mathbf{r}_{ji}/r_{ji}$. We collectively denote the relative atomic geometry by $\mathbf{r} = \{\mathbf{r}_{ji}\}$. An interaction block is written as

$$\tilde{\mathbf{X}} = \mathbf{X}^n + \mathcal{M}\left(\mathrm{Norm}\left(\mathbf{X}^n\right); \mathbf{r}\right),$$
$$\mathbf{X}^{n+1} = \tilde{\mathbf{X}} + \mathcal{F}\left(\mathrm{Norm}\left(\tilde{\mathbf{X}}\right)\right).$$

Here, $\mathcal{M}$ and $\mathcal{F}$ denote the message-passing layer and the node-wise feed-forward layer, respectively. $\mathcal{M}$ exchanges information among neighboring atoms based on their relative geometry, whereas $\mathcal{F}$ operates independently on each node. Norm denotes equivariant merged layer normalization (Liao et al., 2026), in which normalization statistics are shared across all irreducible-representation degrees rather than computed independently for each degree. We evaluate the effect of this interaction block design in Sec. 5.1.

Message-Passing Layer. The message-passing layer performs equivariant message passing using a CGTP followed by neighbor aggregation. Importantly, no nonlinear transformation of the node features is performed on individual edges. Given node features $\mathbf{X}$, the message-passing layer is defined as

$$\mathcal{M}\left(\mathbf{X}; \mathbf{r}\right)_i = \mathrm{Linear}_2\Big[\sum_{j \in \mathcal{N}(i)} \mathbf{w}(r_{ji}) \odot \Big(\mathrm{Linear}_1\left(\mathbf{X}_j\right) \otimes^{SO(3)} \mathbf{Y}(\hat{\mathbf{r}}_{ji})\Big)\Big]. \tag{1}$$

The radial embedding is passed through a multilayer perceptron to generate the tensor-product path weights $\mathbf{w}(r_{ji})$, while the angular embedding $\mathbf{Y}(\hat{\mathbf{r}}_{ji})$ is given by spherical harmonics. The message-passing layer in Eq. 1 is linear with respect to the node features $\mathbf{X}$.

For efficient execution, we use cuEquivariance which provides a fused implementation of operations of the form $\sum_{j \in \mathcal{N}(i)} \mathbf{c}_{ji} \odot \left(\mathbf{A}_j \otimes^{SO(3)} \mathbf{B}_{ji}\right)$, avoiding explicit materialization of intermediate edge-wise tensor-product features. This allows the CGTP operation and neighbor aggregation to be executed jointly in a single kernel. We quantify the resulting throughput and memory gains in Appendix A.5, and evaluate alternative message-passing designs involving additional edge-wise transformations in Sec. 5.2.

Node-wise Feed-Forward Layer. The node-wise feed-forward layer operates independently on each atomic representation and is defined as

$$\mathcal{F}(\mathbf{X})_i = \mathrm{Linear}_2\left[\mathrm{SymmetricContraction}\left(\mathrm{Linear}_1(\mathbf{X}_i)\right)\right].$$

Following the feed-forward design of the original Transformer (Vaswani et al., 2017), Linear_1 expands the channel dimension by a factor of four, and Linear_2 projects it back to the original channel dimension. We use symmetric contraction (Batatia et al., 2022) as an equivariant nonlinear transformation of the node features. We examine the choice of nonlinear transformation and channel expansion ratio in Appendix A.4.

5 Architectural Analysis

We conduct experiments on various architectural designs. Unless otherwise specified, all models are trained on the MPTrj dataset (Deng et al., 2023) for 30 epochs. Following Fu et al. (2025), we randomly sample 5,000 structures from sAlex (Schmidt et al., 2024; Barros-Luque et al., 2026) and use them as the validation set. To isolate the effects of individual architectural components, we use symmetric contraction in the node-wise feed-forward network as the default nonlinearity unless

Table 1: Ablation of interaction block organization. Validation losses are reported as mean ± standard deviation over three random seeds for the EquFlash-S-small configuration and as single-run results for the EquFlash-S-large configuration. ϕ denotes a nonlinear transformation.

Index	Message-passing update	Feed-forward update	Val. loss (Small)↓	Val. loss (Large)↓
1	$\mathbf{X} + \mathcal{M}(\mathrm{Norm}(\mathbf{X}))$	$\tilde{\mathbf{X}} + \mathcal{F}(\mathrm{Norm}(\tilde{\mathbf{X}}))$	$\mathbf{0.212 \pm 0.0004}$	**0.188**
2	$\mathbf{X} + \phi(\mathcal{M}(\mathrm{Norm}(\mathbf{X})))$	$\tilde{\mathbf{X}} + \mathcal{F}(\mathrm{Norm}(\tilde{\mathbf{X}}))$	0.212 ± 0.0022	0.190
3	$\mathbf{X} + \phi(\mathcal{M}(\mathrm{Norm}(\mathbf{X})))$	$\tilde{\mathbf{X}} + \mathrm{Linear}(\mathrm{Norm}(\tilde{\mathbf{X}}))$	0.270 ± 0.0068	0.262
4	$\mathbf{X} + \mathcal{M}(\mathrm{Norm}(\mathbf{X}))$	$\mathcal{F}(\tilde{\mathbf{X}})$	0.226 ± 0.0026	0.210

stated otherwise; the choice of nonlinearity is examined in Appendix A.4. The training objective is $\mathcal{L} = \mathcal{L}_E + \mathcal{L}_F + \mathcal{L}_S$, where $\mathcal{L}_E$ denotes the per-atom energy loss, $\mathcal{L}_F$ the force L2MAE loss, and $\mathcal{L}_S$ the stress loss. Additional training details are provided in Appendix A.10.

5.1 Interaction Block Design

A single tensor-product message-passing layer (Eq. 1) is already used in several equivariant architectures, including NequIP, MACE, and SevenNet, and can likewise benefit from fused implementations. While recent equivariant architectures have increasingly explored more expressive edge-wise processing, the organization of simple CGTP message passing within the interaction block has received comparatively less attention. We therefore revisit how such a message-passing layer should be combined with node-wise feature transformation. We first conduct an ablation study of the interaction block design to examine two design choices: where node-wise nonlinear transformations are applied and whether the node-wise feed-forward update uses a residual connection.

The first row in Table 1 corresponds to the proposed EquFlash-S design, in which the message-passing sub-block contains no node-wise nonlinearity, while nonlinear feature transformation is performed in a separate residual feed-forward sub-block. In the second row, we add a post-aggregation nonlinearity ϕ after message aggregation. In the third row, we replace the feed-forward layer $\mathcal{F}$ with a linear layer, leaving the post-aggregation nonlinearity as the only nonlinear transformation, similar to the design principle of MACE. The final variant adopts a NequIP-inspired design by removing the residual connection from the feed-forward update. For a detailed comparison with CGTP-based architectures, see Appendix A.3.

All variants are implemented within a common EquFlash-S framework, with the remaining architectural and training settings held fixed, [1] allowing us to isolate the effect of the interaction block organization. We evaluate each variant at two model scales, denoted as EquFlash-S-small and EquFlash-S-large, to examine whether the effect of interaction block organization persists as model capacity increases. EquFlash-S-small uses 128 channels, $\ell_{\max} = 3$, and 5 interaction blocks, whereas EquFlash-S-large uses 128 channels, $\ell_{\max} = 4$, and 10 interaction blocks.

As shown in Table 1, adding a nonlinear transformation immediately after message aggregation does not improve the mean validation loss (Row 2), whereas replacing the nonlinear feed-forward layer with a linear layer substantially degrades performance from 0.212 to 0.270. The non-residual feed-forward variant also increases the loss to 0.226. These trends persist at the larger model scale, where EquFlash-S achieves a validation loss of 0.188, compared with 0.190, 0.262, and 0.210 for the three variants, respectively. Taken together, these results suggest that nonlinear feature transformations can be effectively handled by a separate node-wise feed-forward sub-block without requiring an additional post-aggregation nonlinearity in the message-passing sub-block, while the residual feed-forward design plays an important role in maintaining performance. This design remains effective as model capacity increases.

5.2 Message-Passing Layer Design

In this section, we investigate the effect of increasing edge-wise computation within the message-passing layer. We generalize the message-passing layer in Eq. 1 as

$$\mathcal{M}(\mathbf{X}; \mathbf{r})_i = \mathrm{Linear}\Big[\sum_{j \in \mathcal{N}(i)} \mathrm{Conv}_2 \circ \mathrm{Gate} \circ \mathrm{Conv}_1 \left(\mathbf{I}_{ji}\right) \Big],$$

[1] The variant in Row 4 uses a smaller learning rate for stable optimization; see Appendix A.10 for details.

Table 2: Comparison of CGTP- and SO(2)-based message passing layers under architectural modifications and model scaling. $(C, \ell_{\max}, L)$ denotes scale, where C is the number of channels, $\ell_{\max}$ is the maximum rotation order, and L is the number of interaction blocks. Both throughput (atoms/sec) and MaxAtoms were measured during training. MaxAtoms denotes the maximum number of atoms that can be processed on a single GPU. See Appendix A.9 for details on their measurement.

Index	Input	Conv_1	Gate	Conv_2	$(C, \ell_{\max}, L)$	Val. loss↓	Throughput↑	MaxAtoms↑
1	$\mathbf{X}_j$	CGTP	–	–	$(128, 3, 5)$	0.212	19.7k	12.0k
2	$\mathbf{X}_j$	CGTP	–	–	$(128, 3, 10)$	0.201	7.9k	5.2k
3	$\mathbf{X}_j$	CGTP	–	–	$(128, 4, 10)$	**0.188**	2.7k	2.9k
4	$\mathbf{X}_j$	CGTP	✓	–	$(128, 3, 5)$	0.208	3.6k	2.3k
5	$\mathbf{X}_j$	CGTP	✓	CGTP	$(128, 3, 5)$	0.199	2.2k	1.6k
6	$\mathbf{X}_j \Vert \mathbf{X}_i$	CGTP	✓	CGTP	$(128, 3, 5)$	0.192	0.9k	1.0k
7	$\mathbf{X}_j \Vert \mathbf{X}_i$	SO(2)	–	–	$(128, 3, 5)$	0.237	2.7k	2.1k
8	$\mathbf{X}_j \Vert \mathbf{X}_i$	SO(2)	–	–	$(128, 3, 10)$	0.220	1.3k	1.1k
9	$\mathbf{X}_j \Vert \mathbf{X}_i$	SO(2)	–	–	$(128, 4, 10)$	0.198	0.7k	0.6k
10	$\mathbf{X}_j \Vert \mathbf{X}_i$	SO(2)	✓	–	$(128, 3, 5)$	0.223	2.2k	1.8k
11	$\mathbf{X}_j \Vert \mathbf{X}_i$	SO(2)	✓	SO(2)	$(128, 3, 5)$	0.205	1.8k	1.5k

where $\mathbf{I}_{ji} \in \{\mathbf{X}_j, \mathbf{X}_j \Vert \mathbf{X}_i\}$ denotes the input to the edge-wise pipeline, as specified by the Input column in Table 2. Conv_1 and Conv_2 are instantiated using either CGTP- or SO(2)-based equivariant convolutions, and components marked as absent in the table are omitted. The full pipeline, using source–destination features, a gated nonlinearity, and two successive equivariant convolutions, captures the main edge-wise interaction design used in eSEN.

In Table 2, Rows 1, 4, 5, and 6 progressively augment the simple EquFlash-S message-passing layer with components of this edge-wise interaction design: an edge-wise gated nonlinearity, a second equivariant convolution, and source–destination feature mixing. Row 6 therefore represents the full edge-wise pipeline, instantiated with CGTP convolutions. As the edge-wise pipeline becomes more expressive, the validation loss consistently improves from 0.212 to 0.208, 0.199, and 0.192. The throughput decreases from 19.7k to 3.6k, 2.2k, and 0.9k as additional edge-wise transformations are introduced. Similarly, the maximum system size decreases from 12.0k to 2.3k, 1.6k, and 1.0k atoms, respectively.

We next examine an alternative approach: retaining the simple CGTP-based message-passing design of EquFlash-S while increasing model capacity through depth and $\ell_{\max}$ (Rows 1–3). Increasing L from 5 to 10 and then $\ell_{\max}$ from 3 to 4 improves the validation loss from 0.212 to 0.201 and finally 0.188. Notably, the largest model achieves better accuracy than the more elaborate edge-wise CGTP variants while maintaining competitive throughput. These results indicate that, within our setting, allocating additional computation to scaling a simple message-passing architecture yields a more favorable accuracy–efficiency trade-off than increasing the complexity of edge-wise transformations.

Figure 3 further shows that the relative efficiency of the two strategies depends strongly on edge density. For structures with 18 edges per atom, the edge-wise variant (Row 4) is faster than the scaled EquFlash-S model (Row 3). As the number of edges per atom increases to 34, 74, and 158, however, EquFlash-S becomes 1.2×, 1.5×, and 1.6× faster, respectively. This trend reflects the different scaling behavior of the two approaches: the cost of additional edge-wise operations grows with the number of edges, whereas our scaling strategy through additional blocks and higher rotation orders does not incur the same degree of overhead. Consequently, the advantage becomes more pronounced for densely connected atomic graphs, suggesting particular effectiveness in systems with high edge densities.

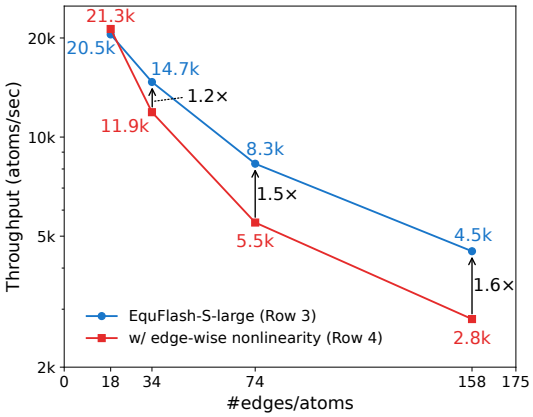

Figure 3: Inference throughput across atomistic systems with different numbers of edges per atom. Comparison of the single-CGTP with (128,4,10) and the CGTP–Gate configurations in Table 2.

The SO(2)-based variants show a different accuracy–efficiency trade-off. We verify our SO(2) implementation against the original eSEN framework under matched architectural and training settings and observe comparable or better validation accuracy in Appendix A.6. Increasing depth and $\ell_{\max}$ improves the validation loss from 0.237 to 0.220 and 0.198 (Rows 7–9), respectively. Alternatively, increasing edge-wise complexity by adding a gated nonlinearity and a second convolution improves the loss to 0.223 and 0.205 (Rows 7, 10, and 11). Notably, the two-convolution variant in Row 11 retains higher throughput than the scaled SO(2) models in Rows 8 and 9. For SO(2)-based models, our implementation lacks a fused kernel for convolution and neighbor aggregation. Whether scaling a simple SO(2)-based architecture would offer similar benefits once such fusion becomes available remains an open question.

The relative advantage of SO(2) becomes more apparent when the message-passing layer contains multiple edge-wise operations. Rows 6 and 11 use analogous edge-wise structures, including a gated nonlinearity and two convolutions, but the SO(2)-based variant achieves 1.8k atoms/s compared with 0.9k atoms/s for CGTP. This comparison suggests that the computational advantage of SO(2) tensor products becomes increasingly relevant as more tensor-product operations are performed per edge. Nevertheless, considering the overall accuracy–efficiency trade-off, we adopt the scaled CGTP design, which achieves the lowest validation loss while retaining higher throughput than most of the more complex edge-wise variants.

6 Evaluation

We evaluate EquFlash-S on established MLIP benchmarks spanning bulk materials, surfaces, and molecular systems. For Matbench Discovery and the MDR Phonon Benchmark (Loew et al., 2025), we denote the three model configurations as EquFlash-S-small, EquFlash-S-medium, and EquFlash-S-large, corresponding to $(C, \ell_{\max}, L) = (128, 3, 5), (128, 3, 10), \text{and } (128, 4, 10)$, respectively, where C is the number of channels, $\ell_{\max}$ is the maximum rotation order, and L is the number of interaction blocks. Additional evaluations on the MatCalc benchmark (Kaplan et al., 2025) are presented in Appendix A.8. Detailed model configurations and training hyperparameters are provided in Appendix A.11.

6.1 Matbench Discovery

Matbench Discovery evaluates MLIPs beyond conventional single-point energy and force prediction. Crystal-structure relaxation is performed on the WBM test set (Wang et al., 2021), which contains 256,963 candidate crystal structures. The relaxed structures are then used to compute the F1 score and structural RMSD. The F1 score evaluates the classification of relaxed structures as stable or unstable with respect to the DFT reference convex hull, while the RMSD quantifies the geometric discrepancy between MLIP-relaxed and DFT-relaxed structures. Lattice thermal conductivity κ is evaluated on 103 structures following the protocol of (Póta et al., 2024). For each structure, finite-displacement calculations are used to obtain the second- and third-order force constants required to solve the Wigner transport equation for κ. The prediction error is summarized by the symmetric relative mean error, denoted by κ_{SRME}, with respect to the reference thermal conductivities. These three metrics are combined into the CPS, which serves as the primary aggregate metric for evaluating overall performance on Matbench Discovery. The benchmark further distinguishes between *compliant* and *non-compliant* settings based on the training data. In the compliant setting, training is restricted to the MPTrj dataset, whereas models trained on additional external data are categorized as non-compliant.

As shown in Tables 3 and 4, EquFlash-S shows consistent accuracy improvements as model size increases while retaining low training cost. In the non-compliant setting, EquFlash-S-large achieves a CPS of 0.907, closely matching the highest score of 0.908 among models trained on the OAM dataset, which combines OMat24 (Barros-Luque et al., 2026), sAlex, and MPTrj, while requiring only 0.85k training GPU hours. Among models with a CPS below 0.900, EquFlash-S-medium and EquFlash-S-small require substantially fewer GPU hours than the others. In the compliant setting, the EquFlash-S variants also demonstrate a favorable accuracy–compute trade-off. In particular, EquFlash-S-large achieves the second-highest CPS of 0.835 with 0.37k GPU hours.

Table 3: Matbench Discovery results of non-compliant models trained on the OAM dataset. GPU hours are reported as training time. † denotes models trained on H20 GPUs; otherwise, H100 GPUs.

Model	Param.	CPS↑	F1↑	κ_{SRME}↓	RMSD↓	GPU hours↓
MatRIS-10M-OAM	10.4M	0.877	0.921	0.218	0.060	5.12k
NequIP-OAM-XL	32.1M	0.886	0.906	0.125	0.063	2.40k
eSEN-30M-OAM	30.2M	0.888	0.925	0.170	0.061	–
TACE-OAM-L	82.9M	0.889	0.910	0.126	0.061	0.84k†
PET-OAM-XL	730M	0.898	0.924	0.119	0.060	204.80k
GRACE-3L-OAM-L	42.1M	0.900	0.925	0.121	0.058	1.98k
EquiformerV3+DeNS-OAM	30.3M	0.902	**0.931**	0.118	0.059	5.67k
TECE-OAM-RRA-1.0	222M	**0.908**	0.929	**0.093**	0.058	7.68k†
EquFlash-S-small	4.5M	0.889	0.915	0.146	0.059	**0.12k**
EquFlash-S-medium	10.3M	0.899	0.923	0.118	0.059	0.33k
EquFlash-S-large	15.5M	0.907	0.929	0.097	**0.057**	0.85k

Table 4: Matbench Discovery results of compliant models trained on MPTrj. GPU hours are reported as training time. * denotes models trained on A100 GPUs; † denotes models trained on H20 GPUs; otherwise, H100 GPUs.

Model	Param.	CPS↑	F1↑	κ_{SRME}↓	RMSD↓	GPU hours↓
SevenNet-l3i5	1.17M	0.714	0.760	0.550	0.085	–
Allegro-MP-L	18.7M	0.720	0.751	0.504	0.082	0.40k
Nequix MP	708k	0.729	0.751	0.446	0.085	0.50k*
NequIP-MP-L	9.6M	0.730	0.761	0.466	0.086	0.29k
Eqnorm MPTrj	1.31M	0.756	0.786	0.408	0.084	2.00k*
MatRIS-L (MatRIS-10M-MP)	10.4M	0.778	0.847	0.489	0.072	5.44k*
eSEN-30M-MP	30.1M	0.797	0.831	0.340	0.075	–
EquiformerV3+DeNS-MP	30.3M	0.830	**0.863**	0.275	0.070	1.07k
DPA-4.0.1-Pro-MPtrj	22.8M	**0.840**	0.857	**0.211**	**0.069**	2.16k†
EquFlash-S-small	4.5M	0.807	0.829	0.285	0.076	**0.06k**
EquFlash-S-medium	10.3M	0.823	0.843	0.252	0.073	0.15k
EquFlash-S-large	15.8M	0.835	0.857	0.240	**0.069**	0.37k

6.2 MDR Phonon Benchmark

The MDR Phonon Benchmark evaluates the accuracy of phonon-property predictions from universal MLIPs across approximately 10,000 non-magnetic semiconductors. Following the benchmark protocol, we evaluate the maximum phonon frequency, $\omega_{\max}$ (K), entropy, S (J K^{-1} mol^{-1}), free energy, F (kJ mol^{-1}), and constant-volume heat capacity, C_V (J K^{-1} mol^{-1}). Structures are first relaxed using the FIRE optimizer with a force-convergence criterion of $f_{\max} = 0.005$ eV Å^{-1}. Phonon properties are then computed using finite displacements of 0.01 Å. As shown in Table 5, EquFlash-S-large achieves the lowest errors across all four phonon properties. The results of models trained on MPTrj are provided in Appendix A.7.

6.3 OC20

The Open Catalyst 2020 (OC20) benchmark (Chanussot et al., 2021) evaluates the prediction of energies and atomic forces for catalyst–adsorbate systems. We compare our results with EquiformerV2 and EquiformerV3 by training on the 2M dataset and predicting total energies rather than adsorption energies. Following the setup in Table 1 of Liao et al. (2026), we predict atomic forces directly instead of computing them as energy gradients. As shown in Table 6, EquFlash-S achieves the lowest energy and force errors among the evaluated models while requiring comparable training GPU hours. However, it should be noted that EquFlash-S has a larger model size than EquiformerV2 and EquiformerV3, and therefore this comparison does not isolate the effect of architectural efficiency.

6.4 SPICE-MACE-OFF

We also evaluate EquFlash-S on the SPICE-MACE-OFF benchmark (Kovács et al., 2025), which covers diverse molecular systems. As shown in Table 7, EquFlash-S achieves competitive performance across seven test subsets, including strong results on PubChem and solvated amino acids (Sol. AA), although it exhibits higher errors than eSEN on DES370K and Dipeptides.

Table 5: MDR Phonon Benchmark results of models trained on the OAM dataset.

Model	Param.	$\omega_{\max}\downarrow$	$S\downarrow$	$F\downarrow$	$C_V\downarrow$
EquiformerV2-M-OAM	31.2M	50	25	7	9
Orb-V3	25.5M	19	16	6	4
GRACE-2L-OAM	12.6M	19	14	5	4
SevenNet-MF-ompa	25.7M	15	8	3	3
eSEN-30M-OAM	30.2M	15	10	4	3
MatRIS-10M-OAM	10.4M	8	9	3	2
EquiformerV3+DeNS-OAM	30.3M	8.8	7.5	2.1	2.5
EquFlash-S-large	15.5M	**5.9**	**5.3**	**1.8**	**1.3**

Table 6: OC20-2M results. Errors are reported as the average across the four subsplits of the validation set. GPU hours represent the training time for each model using H100 GPUs.

Model	Param.	Energy MAE↓	Force MAE↓	GPU hours↓
EquiformerV2	54M	242	19.73	270
EquiformerV3	91M	201	18.15	**171**
EquFlash-S	228M	**185.7**	**17.35**	205

Table 7: SPICE-MACE-OFF results. Errors are reported on the test set. E and F denote the mean absolute errors (MAEs) of energy (meV/atom) and force (meV/Å), respectively.

Dataset	MACE-4.7M		EScAIP-45M		eSEN-3.2M		eSEN-6.5M		EquFlash-S-3.0M		EquFlash-S-6.6M	
	E	F	E	F	E	F	E	F	E	F	E	F
PubChem	0.88	14.75	0.53	5.86	0.22	6.10	**0.15**	4.21	0.20	5.06	0.16	**3.71**
DES370K M.	0.59	6.58	0.41	3.48	0.17	1.85	**0.13**	**1.24**	0.20	6.27	0.17	4.81
DES370K D.	0.54	6.62	0.38	2.18	0.20	2.77	**0.15**	**2.12**	0.23	5.79	0.19	4.43
Dipeptides	0.42	10.19	0.31	5.21	0.10	3.04	**0.07**	**2.00**	0.20	4.81	0.16	3.34
Sol. AA	0.98	19.43	0.61	11.52	0.30	5.76	0.25	3.68	0.17	4.16	**0.12**	**2.75**
Water	0.83	13.57	0.72	10.31	0.24	3.88	0.15	**2.50**	0.13	3.89	**0.11**	2.64
QMugs	0.45	16.93	0.41	8.74	0.16	5.70	**0.12**	3.78	0.24	4.32	0.15	**2.94**

7 Conclusion

We revisited two fundamental design choices for equivariant MLIPs: how to organize interaction blocks and how much computation to perform on edges. Based on our findings, we developed EquFlash-S, which combines a residual message-passing sub-block with a separate residual nonlinear node-wise feed-forward sub-block. By using a single CGTP for message passing, EquFlash-S keeps edge-wise computation lightweight and enables fused kernels to jointly perform tensor products and neighbor aggregation without materializing intermediate high-order edge features.

Our experiments show that increasing edge-wise expressivity can improve accuracy, but at a substantial computational and memory cost. In contrast, scaling the capacity of the simple, kernel-friendly architecture achieves a better accuracy–efficiency trade-off. Across Matbench Discovery, the MDR Phonon Benchmark, MatCalc, and OC20, EquFlash-S achieves competitive performance while maintaining high computational efficiency. These results highlight the importance of jointly considering model architecture and kernel implementation when designing efficient equivariant architectures: rather than increasing the complexity of edge-wise operations, keeping the edge operator simple can enable efficient scaling of model capacity.

Limitations Our architectural ablations are conducted on MPTrj under a limited set of model scales and training settings, and the edge-heavy alternatives are not evaluated as fully scaled model families. In addition, throughput and memory comparisons between architectural variants are measured within our implementation rather than across independently optimized implementations. On benchmarks with lower edge density, arising from a maximum-neighbor constraint in OC20 or fewer neighbors per atom in SPICE-MACE-OFF, we observed less pronounced throughput gains than on the bulk structures used for our Matbench Discovery evaluation. The benefits of our design are therefore most pronounced for systems with high edge density, where edge-wise computation constitutes a larger fraction of the overall computational cost.

AI USE STATEMENT

In this work, we used generative AI tools for translation, implementing methods, and providing feedback on experiments. We did not use generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, propose or refine hypotheses, design on research methodology or experiments, provide feedback on research methodology, clean or reformat datasets, support qualitative or thematic data analysis, or interpret results. Formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting in the writing of proofs were not applicable to this work.

Additionally, we used generative AI tools to create or edit software code and to draft parts of the manuscript. All AI-assisted work was reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of our model architecture and experimental configurations to facilitate reproducibility. Section 4 and Appendix A.1 describe the EquFlash-S architecture and its implementation details. Sections 5 and 6 present the experimental setups for the architectural analyses and benchmark evaluations. Appendix A.9 specifies the hardware and measurement procedures used for throughput and memory evaluations, while Appendices A.10 and A.11 provide the training configurations and hyperparameters for the architectural analyses and benchmark evaluations, respectively. An anonymized implementation of EquFlash-S is provided in the supplementary material to further facilitate reproducibility.

REFERENCES

Brandon Anderson, Truong Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. In *Advances in neural information processing systems*, 2019.

Luis Barros-Luque, Muhammed Shuaibi, Xiang Fu, Brandon M Wood, Misko Dzamba, Meng Gao, Ammar Rizvi, Matt Uyttendaele, C Lawrence Zitnick, and Zachary W Ulissi. The Open Materials 2024 (OMat24) inorganic materials dataset and models. *Nature Computational Science*, pp. 1–11, 2026.

Ilyes Batatia, David P Kovacs, Gregor Simm, Christoph Ortner, and Gábor Csányi. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. In *Advances in Neural Information Processing Systems*, 2022.

Ilyes Batatia, Philipp Benner, Yuan Chiang, Alin M Elena, Dávid P Kovács, Janosh Riebesell, Xavier R Advincula, Mark Asta, Matthew Avaylon, William J Baldwin, et al. A foundation model for atomistic materials chemistry. *The Journal of chemical physics*, 163(18), 2025.

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, 2022.

Vivek Bharadwaj, Austin Glover, Aydın Buluç, and James Demmel. An efficient sparse kernel generator for O(3)-equivariant deep networks. In *SIAM Conference on Applied and Computational Discrete Algorithms*, 2025.

Lowik Chanussot, Abhishek Das, Siddharth Goyal, Thibaut Lavril, Muhammed Shuaibi, Morgane Riviere, Kevin Tran, Javier Heras-Domingo, Caleb Ho, Weihua Hu, et al. Open catalyst 2020 (OC20) dataset and community challenges. *Acs Catalysis*, 11(10):6059–6072, 2021.

Yuan Chiang, Tobias Kreiman, Christine Zhang, Matthew Kuner, Elizabeth Weaver, Ishan Amin, Hyunsoo Park, Yunsung Lim, Jihan Kim, Daryl Chrzan, et al. MLIP Arena: Advancing fairness and transparency in machine learning interatomic potentials via an open, accessible benchmark platform. In *Advances in Neural Information Processing Systems*, 2026.

Stefan Chmiela, Alexandre Tkatchenko, Huziel E Sauceda, Igor Poltavsky, Kristof T Schütt, and Klaus-Robert Müller. Machine learning of accurate energy-conserving molecular force fields. *Science advances*, 3(5):e1603015, 2017.

Bowen Deng, Peichen Zhong, KyuJung Jun, Janosh Riebesell, Kevin Han, Christopher J Bartel, and Gerbrand Ceder. CHGNet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nature Machine Intelligence*, 5(9):1031–1041, 2023.

Peter Eastman, Pavan Kumar Behara, David L Dotson, Raimondas Galvelis, John E Herr, Josh T Horton, Yuezhi Mao, John D Chodera, Benjamin P Pritchard, Yuanqing Wang, et al. Spice, a dataset of drug-like molecules and peptides for training machine learning potentials. *Scientific Data*, 10(1):11, 2023.

Xiang Fu, Brandon M Wood, Luis Barroso-Luque, Daniel S. Levine, Meng Gao, Misko Dzamba, and C. Lawrence Zitnick. Learning smooth and expressive interatomic potentials for physical property prediction. In *International Conference on Machine Learning*, 2025.

Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in neural information processing systems*, 33:1970–1981, 2020.

Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2020.

Johannes Gasteiger, Florian Becker, and Stephan Günnemann. GemNet: Universal directional graph neural networks for molecules. In *Advances in Neural Information Processing Systems*, 2021.

Aaron D Kaplan, Runze Liu, Ji Qi, Tsz Wai Ko, Bowen Deng, Janosh Riebesell, Gerbrand Ceder, Kristin A Persson, and Shyue Ping Ong. A foundational potential energy surface dataset for materials. *arXiv preprint arXiv:2503.04070*, 2025.

Geonu Kim, Byunggook Na, Gunhee Kim, Hyuntae Cho, Seungjin Kang, Hee Sun Lee, Saerom Choi, Heejae Kim, Seungwon Lee, and Yongdeok Kim. Benchmark of machine learning force fields for semiconductor simulations: datasets, metrics, and comparative analysis. In *Advances in Neural Information Processing Systems*, 2023.

Dávid Péter Kovács, J Harry Moore, Nicholas J Browning, Ilyes Batatia, Joshua T Horton, Yixuan Pu, Venkat Kapil, William C Witt, Ioan-Bogdan Magdau, Daniel J Cole, et al. MACE-OFF: Short-range transferable machine learning force fields for organic molecules. *Journal of the American Chemical Society*, 147(21):17598–17611, 2025.

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 *International Conference on Machine Learning*, 2025.

Daniel S Levine, Muhammed Shuaibi, Evan Walter Clark Spotte-Smith, Michael G Taylor, Muhammad R Hasyim, Kyle Michel, Ilyes Batatia, Gábor Csányi, Misko Dzamba, Peter Eastman, et al. The Open Molecules 2025 (OMol25) dataset, evaluations, and models. *arXiv preprint arXiv:2505.08762*, 2025.

Yi-Lun Liao, Brandon Wood, Abhishek Das, and Tess Smidt. Equiformerv2: Improved equivariant transformer for scaling to higher-degree representations. In *International Conference on Learning Representations*, 2024.

Yi-Lun Liao, Alexander J Hoffman, Sabrina C Shen, Alexandre Duval, Sam Walton Norwood, and Tess Smidt. Equiformerv3: Scaling efficient, expressive, and general SE(3)-equivariant graph attention transformers. *arXiv preprint arXiv:2604.09130*, 2026.

Antoine Loew, Dewen Sun, Hai-Chen Wang, Silvana Botti, and Miguel AL Marques. Universal machine learning interatomic potentials are ready for phonons. *npj Computational Materials*, 11 (1):178, 2025.

Arslan Mazitov, Filippo Bigi, Matthias Kellner, Paolo Pegolo, Davide Tisi, Guillaume Fraux, Sergey Pozdnyakov, Philip Loche, and Michele Ceriotti. PET-MAD as a lightweight universal interatomic potential for advanced materials modeling. *Nature Communications*, 16(1):10653, 2025.

Jinuk Moon, Uchan Jeon, Seokhyun Choung, and Jeong Woo Han. Catbench framework for benchmarking machine learning interatomic potentials in adsorption energy predictions for heterogeneous catalysis. *Cell Reports Physical Science*, 6(12), 2025.

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, 2023.

Mark Neumann, James Gin, Benjamin Rhodes, Steven Bennett, Zhiyi Li, Hitarth Choubisa, Arthur Hussey, and Jonathan Godwin. Orb: A fast, scalable neural network potential. *arXiv preprint arXiv:2410.22570*, 2024.

NVIDIA. cuEquivariance: High-performance equivariant neural network operations, 2025. URL `https://github.com/NVIDIA/cuEquivariance`.

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, 2024.

Saro Passaro and C Lawrence Zitnick. Reducing SO(3) convolutions to SO(2) for efficient equivariant GNNs. In *International Conference on Machine Learning*, 2023.

Balázs Póta, Paramvir Ahlawat, Gábor Csányi, and Michele Simoncelli. Thermal conductivity predictions with foundation atomistic models. *arXiv preprint arXiv:2408.00755*, 2024.

Sergey Pozdnyakov and Michele Ceriotti. Smooth, exact rotational symmetrization for deep learning on point clouds. In *Advances in Neural Information Processing Systems*, 2023.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):140022, 2014.

Benjamin Rhodes, Sander Vandenhaute, Vaidotas Šimkus, James Gin, Jonathan Godwin, Tim Duignan, and Mark Neumann. Orb-v3: atomistic simulation at scale. *arXiv preprint arXiv:2504.06231*, 2025.

Janosh Riebesell, Rhys EA Goodall, Philipp Benner, Yuan Chiang, Bowen Deng, Gerbrand Ceder, Mark Asta, Alpha A Lee, Anubhav Jain, and Kristin A Persson. A framework to evaluate machine learning crystal stability predictions. *Nature Machine Intelligence*, 7(6):836–847, 2025.

Sushree Jagriti Sahoo, Mikael Maraschin, Zachary Ward Ulissi, C Lawrence Zitnick, Joel B Varley, Joseph Gauthier, Nitish Govindarajan, and Muhammed Shuaibi. The Open Catalyst 2025 (OC25) dataset and models for solid-liquid interfaces. In *AI4X–Accelerate Conference 2026*, 2025.

Jonathan Schmidt, Tiago FT Cerqueira, Aldo H Romero, Antoine Loew, Fabian Jäger, Hai-Chen Wang, Silvana Botti, and Miguel AL Marques. Improving machine-learning models in materials science through large datasets. *Materials Today Physics*, 48:101560, 2024.

Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in neural information processing systems*, 2017.

Justin S Smith, Olexandr Isayev, and Adrian E Roitberg. Ani-1, a data set of 20 million calculated off-equilibrium conformations for organic molecules. *Scientific data*, 4(1):170193, 2017.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Richard Tran, Janice Lan, Muhammed Shuaibi, Brandon M Wood, Siddharth Goyal, Abhishek Das, Javier Heras-Domingo, Adeesh Kolluru, Ammar Rizvi, Nima Shoghi, et al. The Open Catalyst 2022 (OC22) dataset and challenges for oxide electrocatalysts. *Acs Catalysis*, 13(5):3066–3084, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, 2017.

Hai-Chen Wang, Silvana Botti, and Miguel AL Marques. Predicting stable crystalline compounds using chemical similarity. *npj Computational Materials*, 7(1):12, 2021.

Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. In *Advances in Neural information processing systems*, 2018.

Brandon Wood, Misko Dzamba, Xiang Fu, Meng Gao, Muhammed Shuaibi, Luis Barroso-Luque, Kareem Abdelmaqsoud, Vahe Gharakhanyan, John Kitchin, Daniel Levine, et al. UMA: A family of universal models for atoms. In *Advances in Neural Information Processing Systems*, 2026.

Zemin Xu, Wenbo Xie, and P Hu. Edge cluster expansion with radial rotary attention for interatomic potentials. *arXiv preprint arXiv:2607.10664*, 2026.

Yuanchang Zhou, Siyu Hu, Xiangyu Zhang, Hongyu Wang, Guangming Tan, and Weile Jia. MatRIS: Toward reliable and efficient pretrained machine learning interatomic potentials. In *International Conference on Learning Representations*, 2026.

A Appendix

A.1 Further Architectural Details

Input Embedding. Given an atomic configuration with atomic numbers Z_i and positions $\mathbf{r}_i$, we construct a local neighborhood graph using a cutoff radius r_{cut}. Each atom is assigned a learnable embedding based on its atomic number,

$$\mathbf{X}_i^0 = \text{Embed}(Z_i),$$

where the initial node features consist of invariant scalar ($0e$) representations.

For each neighboring pair of atoms, let $\mathbf{r}_{ji}$ denote the relative displacement from atom j to atom i, with distance $r_{ji} = \|\mathbf{r}_{ji}\|$ and unit direction $\hat{\mathbf{r}}_{ji} = \mathbf{r}_{ji}/r_{ji}$.

The interatomic distance is expanded using Bessel basis functions multiplied by an XPLOR cutoff function:

$$\mathbf{b}(r_{ji}) = f_{\text{cut}}(r_{ji}) \left[B_1(r_{ji}), \ldots, B_{N_b}(r_{ji})\right].$$

The radial features are passed through a multilayer perceptron to obtain the distance-dependent tensor-product weights:

$$\mathbf{w}(r_{ji}) = \text{MLP}(\mathbf{b}(r_{ji})).$$

Angular edge features are constructed by evaluating spherical harmonics along the interatomic direction:

$$\mathbf{Y}(\hat{\mathbf{r}}_{ji}) = \bigoplus_{\ell=0}^{\ell_{\max}} \mathbf{Y}^{(\ell)}(\hat{\mathbf{r}}_{ji}),$$

where $\mathbf{Y}^{(\ell)}$ denotes the real spherical harmonics of degree ℓ.

Output Head. After N interaction blocks, the final node representations contain features of multiple irreducible representations. A linear projection maps these representations to a single invariant scalar ($0e$) feature for each atom, yielding the normalized atomic energy contribution E_i. The total energy is computed by applying a global energy scale s and an element-dependent energy shift μ_{Z_i}:

$$E = \sum_i \left(sE_i + \mu_{Z_i}\right).$$

The force on atom i is given by

$$\mathbf{f}_i = -\frac{\partial E}{\partial \mathbf{r}_i},$$

which ensures that the predicted forces form a conservative force field. Alternatively, for direct force prediction, we extract the $1o$ vector features $\mathbf{X}_i^{(1o)}$ from each final atomic representation and apply an equivariant linear projection to obtain the predicted force:

$$\mathbf{f}_i^{\text{direct}} = \text{Linear}_{1o\to 1o}\left(\mathbf{X}_i^{(1o)}\right).$$

Unlike energy-gradient forces, directly predicted forces are not guaranteed to form a conservative force field.

For a periodic cell of volume V, the stress tensor is obtained from the derivative of the energy with respect to the strain tensor $\boldsymbol{\varepsilon}$:

$$\boldsymbol{\sigma} = \frac{1}{V}\frac{\partial E}{\partial \boldsymbol{\varepsilon}}.$$

A.2 Additional Related Work

Invariant Atomistic Networks. Early atomistic neural networks encoded geometric information primarily through rotation-invariant scalar features. SchNet (Schütt et al., 2017) constructs continuous-filter convolutions from pairwise interatomic distances, while DimeNet (Gasteiger et al., 2020) augments distance-based representations with bond-angle information through directional message passing. GemNet (Gasteiger et al., 2021) further extends this framework by incorporating richer geometric interactions, including dihedral information. More recently, MatRIS (Zhou et al., 2026) models three-body interactions using a linear-complexity separable attention mechanism, demonstrating that carefully designed invariant architectures can achieve accuracy competitive with leading equivariant models at substantially lower training cost. These models obtain efficient and expressive representations by constructing geometric features that are invariant under rotations, although directional information is not explicitly maintained in their intermediate features.

Relaxing Symmetry Constraints. One approach is to remove strict equivariance from the architecture and instead encourage rotational symmetry during training or correct symmetry violations at the output level. PET (Pozdnyakov & Ceriotti, 2023) directly processes Cartesian point and edge features using conventional transformer blocks, with rotational symmetry learned through data augmentation; PET-MAD (Mazitov et al., 2025) extends this design to a broadly applicable interatomic potential. Orb (Neumann et al., 2024) similarly adopts an unconstrained message-passing architecture and improves molecular-dynamics stability by removing residual net forces and torques. Orb-v3 (Rhodes et al., 2025) further introduces EquiGrad, an auxiliary objective that encourages rotational invariance of the predicted energy, while also exploring direct force prediction and sparse graph construction. These approaches permit the use of flexible and highly optimized neural-network components, but rotational symmetry is learned, regularized, or corrected rather than guaranteed for every input.

Datasets. The development of MLIPs has been supported by increasingly large and diverse atomistic datasets. For molecular systems, QM9 (Ramakrishnan et al., 2014), ANI-1 (Smith et al., 2017), and MD17 (Chmiela et al., 2017) have been widely used to develop and evaluate molecular representations and interatomic potentials. More recently, SPICE (Eastman et al., 2023) and OMol25 (Levine et al., 2025) have expanded the coverage of molecular configurations and chemical environments.

For inorganic materials, MPTrj (Deng et al., 2023), Alexandria (Schmidt et al., 2024), and OMat24 (Barros-Luque et al., 2026) provide large-scale DFT calculations across diverse compositions and atomic configurations, facilitating the development of universal MLIPs. MatPES (Kaplan et al., 2025) emphasizes carefully sampled configurations and high-quality DFT references, while the MAD dataset (Mazitov et al., 2025) used to train PET-MAD covers both inorganic and organic materials with diverse and highly distorted atomic configurations. For catalytic systems, OC20 (Chanussot et al., 2021) established a large-scale dataset of catalyst–adsorbate structures, while OC22 (Tran et al., 2023) extended the coverage to oxide electrocatalysts. More recently, OC25 (Sahoo et al., 2025) introduced diverse solid–liquid interfaces with explicit solvent environments.

Benchmarks. As MLIPs are increasingly applied to practical atomistic simulations, their evaluation has expanded beyond conventional energy and force prediction errors. SAMD23 (Kim et al., 2023) introduced semiconductor datasets and evaluation metrics for assessing both predictive accuracy and simulation behavior. Matbench Discovery (Riebesell et al., 2025) assesses materials discovery through stability classification, structural relaxation, and thermal conductivity predictions. The MDR Phonon Benchmark (Loew et al., 2025) evaluates phonon-related thermodynamic properties, while CatBench (Moon et al., 2025) examines adsorption energy predictions for heterogeneous catalysis. More broadly, MLIP Arena (Chiang et al., 2026) provides an evaluation platform covering physical consistency, chemical reactivity, and stability under extreme conditions.

A.3 Comparison of Equivariant Interaction Designs

Several models closely related to EquFlash-S are reported on the Matbench Discovery leaderboard. To clarify their relationship to our architecture, we compare their interaction designs alongside those of representative equivariant MLIPs.

At the level of abstraction considered here, NequIP and SevenNet combine aggregated CGTP messages with input node features through a projected residual connection, followed by a node-wise gated nonlinearity. MACE also employs a projected residual connection but instead applies symmetric contraction to the aggregated messages, introducing higher-order nonlinear interactions before the residual addition. In contrast, EquFlash-S separates CGTP message passing and nonlinear node-wise feature transformation into two residual sub-blocks. These distinctions motivate the controlled architectural comparisons presented in Sec. 5.1.

Let $\mathbf{X}_i^n$ denote the feature of atom i entering the n-th interaction block. Although these architectures differ in their detailed parameterizations and implementations, their interaction blocks can be summarized at a common level of abstraction as follows.

NequIP and SevenNet:

$$\mathbf{m}_{ji} = \mathbf{w}(r_{ji}) \odot \left(\text{Linear}\left(\mathbf{X}_j^n\right) \otimes^{SO(3)} \mathbf{Y}(\hat{\mathbf{r}}_{ji})\right)$$

$$\tilde{\mathbf{X}}_i = \text{Linear}\left(\mathbf{X}_i^n\right) + \text{Linear}\left[\sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ji}\right]$$

$$\mathbf{X}_i^{n+1} = \text{Gate}\left(\tilde{\mathbf{X}}_i\right)$$

MACE:

$$\mathbf{m}_{ji} = \mathbf{w}(r_{ji}) \odot \left(\text{Linear}\left(\mathbf{X}_j^n\right) \otimes^{SO(3)} \mathbf{Y}(\hat{\mathbf{r}}_{ji})\right)$$

$$\mathbf{X}_i^{n+1} = \text{Linear}\left(\mathbf{X}_i^n\right) + \text{SymmetricContraction}\left[\sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ji}\right]$$

EquFlash-S:

$$\mathbf{m}_{ji} = \mathbf{w}(r_{ji}) \odot \left(\text{Linear}\left(\text{Norm}\left(\mathbf{X}_j\right)\right) \otimes^{SO(3)} \mathbf{Y}(\hat{\mathbf{r}}_{ji})\right)$$

$$\tilde{\mathbf{X}}_i = \mathbf{X}_i + \text{Linear}\left[\sum_{j \in \mathcal{N}(i)} \mathbf{m_{ji}}\right]$$

$$\mathbf{X}_i^{n+1} = \tilde{\mathbf{X}}_i + \mathcal{F}\left(\text{Norm}\left(\tilde{\mathbf{X}}_i\right)\right)$$

A.4 NODE-WISE FEED-FORWARD DESIGN

We examine the design of the node-wise feed-forward network. We first compare several equivariant nonlinear transformations, including the identity mapping, gated activation, a bilinear gated activation, and symmetric contraction. The bilinear gate can be viewed as an equivariant variant of a gated linear unit, where invariant scalar features modulate a bilinear equivariant transformation,

$$\mathrm{BilinearGate}(\mathbf{X}) = \sigma(\mathbf{X}^{0e}) \odot \left(\mathrm{Linear}_1\left(\mathbf{X}\right) \otimes^{SO(3)} \mathrm{Linear}_2\left(\mathbf{X}\right) \right).$$

For symmetric contraction, we use correlation degree two. We additionally vary the intermediate channel expansion ratio of the feed-forward network while fixing symmetric contraction as the nonlinearity.

As shown in Table 8, removing the nonlinear transformation substantially degrades accuracy, with the identity mapping yielding a mean validation loss of 0.285. Among the nonlinear variants at an expansion ratio of one, symmetric contraction achieves the lowest loss of 0.220, compared with 0.231 for the conventional gate and 0.223 for the bilinear gate. These results support the importance of expressive nonlinear processing in the node-wise feed-forward stage.

We then increase the intermediate channel width while using symmetric contraction. Increasing the expansion ratio from $1\times$ to $2\times$ and $4\times$ progressively reduces the mean validation loss from 0.220 to 0.216 and 0.212, respectively. Increasing the ratio further to $8\times$ yields only a marginal reduction to 0.211, while decreasing throughput from 19.7k to 17.8k atoms/s and reducing the maximum system size from 12.0k to 10.5k atoms compared with the $4\times$ configuration. We therefore use a $4\times$ intermediate channel expansion with symmetric contraction in the final EquFlash-S architecture, providing a favorable balance between accuracy and computational efficiency.

Table 8: Ablation of the node-wise feed-forward network. The expansion-ratio study uses symmetric contraction as the nonlinear transformation. Reported validation losses are the mean $\pm$ standard deviation of the best validation loss over 30 epochs across three random seeds. Both throughput (atoms/sec) and MaxAtoms were measured during training. MaxAtoms denotes the maximum number of atoms that can be processed on a single GPU. See Appendix A.9 for details on their measurement.

Nonlinearity	Expansion ratio	Valid Loss↓	Throughput↑	MaxAtoms↑
Identity	1	0.285 ± 0.0040	22.4k	13.5k
Gate	1	0.231 ± 0.0017	21.8k	13.0k
Bilinear Gate	1	0.223 ± 0.0004	18.9k	12.0k
Symmetric Contraction	1	0.220 ± 0.0010	22.2k	13.5k
Symmetric Contraction	2	0.216 ± 0.0004	21.6k	13.0k
Symmetric Contraction	4	0.212 ± 0.0004	19.7k	12.0k
Symmetric Contraction	8	$\mathbf{0.211 \pm 0.0010}$	17.8k	10.5k

A.5 cuEQUIVARIANCE VS. E3NN

Table 9 evaluates the computational benefits of progressively applying kernel-level optimizations to EquFlash-S. Replacing the e3nn implementation with cuEquivariance substantially improves both training and inference throughput, from 1.3k to 11.7k atoms/s and from 5.0k to 33.3k atoms/s, respectively. Fusing the scatter–gather operations with the tensor product further increases throughput to 19.7k atoms/s during training and 57.4k atoms/s during inference, while substantially increasing the maximum supported system size. Finally, applying `torch.compile` further improves throughput to 28.6k atoms/s and 74.3k atoms/s for training and inference, respectively. Overall, these results demonstrate that the simplified message-passing design of EquFlash-S can effectively benefit from optimized equivariant kernels, operator fusion, and compilation.

Table 9: Throughput (atoms/sec) and MaxAtoms with progressive kernel optimizations. MaxAtoms denotes the maximum number of atoms that can be processed on a single GPU. See Appendix A.9 for details on their measurement.

Implementation	Training		Inference	
	Throughput	MaxAtoms	Throughput	MaxAtoms
e3nn	1.3k	1.9k	5.0k	3.8k
+ cuEquivariance (CGTP)	11.7k	3.2k	33.3k	6.2k
+ cuEquivariance (aggregation fusion)	19.7k	12.0k	57.4k	26.0k
+ `torch.compile`	28.6k	12.0k	74.3k	31.0k

A.6 VALIDATION OF THE SO(2) IMPLEMENTATION

To validate our implementation of the SO(2)-based convolution, we compare it against the original eSEN implementation under matched settings. For both implementations, we use 128 channels, $\ell_{\max} = 3$, and five interaction layers. We adopt the loss function and learning-rate scheduler used in eSEN, train both implementations for 30 epochs, and set the maximum number of atoms per batch to 4,000. We evaluate three learning rates to account for differences in optimization behavior. As shown in Figure 4, although the training trajectories differ, our implementation achieves comparable or lower validation error than the original eSEN implementation across the tested learning rates. These results indicate that our SO(2) implementation provides a sufficiently strong baseline for the ablation experiments in the main text.

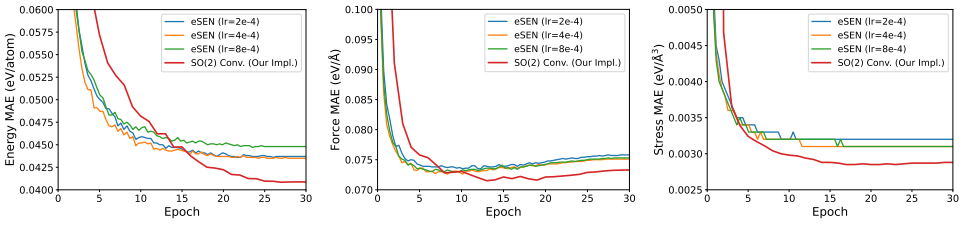

Figure 4: Comparison of the original eSEN implementation and our SO(2) implementation under matched settings with 128 channels, $\ell_{\max} = 3$, and five interaction layers. Both models use the eSEN loss function and learning-rate scheduler and are trained for 30 epochs with a maximum of 4,000 atoms per batch. Results are shown for three learning rates.

A.7 MDR Phonon Benchmark Results of MPTrj-trained Models

Table 10: MDR Phonon Benchmark results of models trained on the MPTrj dataset.

Model	Param.	$\omega_{\max}\downarrow$	$S\downarrow$	$F\downarrow$	$C_V\downarrow$
M3GNet	0.2M	98	150	56	22
CHGNet	0.4M	89	114	45	21
MACE	4.7M	61	60	24	13
GRACE-2L-MPtrj	15.3M	40	25	28	13
SevenNet-l3i5	1.2M	26	28	10	5
eSEN-30M-MP	30.1M	21	**13**	**5**	4
MatRIS-L	10.4M	28	30	12	7
EquFlash-S-large	15.8M	**17.6**	**12.8**	**4.7**	**3.1**

A.8 MatCalc Benchmark Results

MatCalc evaluates MLIPs on equilibrium and near-equilibrium material properties, including structural, energetic, elastic, and vibrational properties. As shown in Table 11, EquFlash-S consistently achieves competitive performance across the MatPES-, MPTrj-, and OAM-trained settings, with particularly strong results on near-equilibrium properties.

Table 11: MatCalc results.

Model	Param.	Equilibrium		Near-equilibrium			
		$d\downarrow$	$E_f\downarrow$	$K\downarrow$	$G\downarrow$	$CV\downarrow$	$f/f_{\mathrm{DFT}}\uparrow$
MatPES-trained models							
M3GNet	0.66M	0.42	0.11	26	25	27	**0.97**
CHGNet	2.7M	0.43	0.082	24	21	23	0.91
TensorNet	0.84M	0.37	0.081	18	15	13	0.93
MatRIS	1.4M	0.54	0.068	**15**	**13**	16	0.96
EquFlash-S	1.4M	**0.28**	**0.064**	16	15	**11**	0.95
MPTrj-trained models							
CHGNet	2.7M	0.51	0.092	17.0	30.0	24.0	0.830
MACE-L	5.7M	0.43	–	25.3	22.5	11.6	0.829
SevenNet-l3i5	1.17M	0.55	0.057	13.2	170.2	8.03	0.922
EquiformerV2-S-DeNS	31.2M	**0.25**	**0.033**	27.2	29.1	25.9	0.964
eSEN-30M-MP	30.1M	0.34	0.039	18.8	77.3	**4.66**	**0.986**
MatRIS-M	6.3M	0.32	0.041	**12.4**	16.4	7.39	0.983
EquFlash-S-large	15.8M	0.30	0.035	12.5	**16.1**	4.88	0.978
OAM-trained models							
SevenNet-MF-ompa	25.7M	0.502	0.028	13.3	32.2	4.60	0.976
EquiformerV2-M-OAM	86.6M	**0.235**	**0.017**	25.4	17.5	80.4	**0.999**
eSEN-30M-OAM	30.2M	0.299	0.089	11.9	14.8	4.35	0.996
MatRIS-10M-OAM	10.4M	0.316	0.025	10.6	**13.3**	3.97	0.985
EquFlash-S-large	15.5M	0.237	0.024	**10.5**	14.8	**3.77**	0.989

A.9 THROUGHPUT MEASUREMENT

All throughput measurements were conducted on an H100 GPU (80 GB). For Tables 2, 8, and 9, and Figures 1 (left panel) and 3, throughput was measured for atomistic systems containing 200, 500, 1,000, 2,000, 3,000, 4,000, 8,000, and 12,000 atoms, and the best throughput across these system sizes is reported. $n_{\text{edge}}/n_{\text{atom}}$ was set to approximately 45, except for Figure 3, where n_{edge} and n_{atom} were varied independently. For each measurement, the first 10 iterations were used as warm-up runs and excluded from timing, after which the runtime was averaged over 50 iterations. MaxAtoms was determined separately using a finer-grained search over system sizes. For Figure 1 (right panel), inference time and throughput were measured for atomistic systems containing 256, 512, 1,024, 2,048, and 4,096 atoms ($n_{\text{edge}}/n_{\text{atom}} \approx 18$), and the best throughput across these system sizes is reported. See Tables 12, 13, 14, 15, and 16 for the results for each system size. The first 10 iterations were excluded as warm-up runs, and the runtime was averaged over the subsequent 20 iterations. The results in Figure 1 were obtained using both cuEquivariance and `torch.compile`, whereas those in Tables 2 and 8 and Figure 3 were obtained using cuEquivariance only.

Table 12: Inference time (ms) and throughput (atoms/sec) of various models for atomistic systems containing 256 atoms. Values in parentheses denote the forward time (including force computation via energy gradients), while values outside parentheses denote the end-to-end time including the forward time and graph construction time.

Model	Acceleration	Time	Throughput
Prophet-OAME-MBD	openEquivariance	273.83 (265.94)	0.9k
TECE-OAM-RRA-1.0	OpenEquivariance, AOTInductor	72.52 (70.31)	3.5k
EquiformerV3+DeNS-OAM	-	99.41 (91.98)	2.6k
GRACE-3L-OAM-L	XLA JIT	45.96 (35.27)	5.6k
PET-OAM-XL	TorchScript	95.15 (68.48)	2.7k
TACE-OAM-L	OpenEquivariance, AOTInductor	20.66 (18.18)	12.4k
eSEN-30M-OAM	-	112.81 (105.44)	2.3k
NequIP-OAM-XL	OpenEquivariance, AOTInductor	8.08 (6.71)	31.7k
MatRIS-10M-OAM	-	182.23 (118.83)	1.4k
SevenNet-Omni-i12	cuEquivariance	96.60 (92.62)	2.7k
EquFlash-S-small	cuEquivariance, `torch.compile`	48.44 (33.37)	5.3k
EquFlash-S-medium	cuEquivariance, `torch.compile`	62.00 (52.10)	4.1k
EquFlash-S-large	cuEquivariance, `torch.compile`	91.58 (85.39)	2.8k

Table 13: Inference time (ms) and throughput (atoms/sec) of various models for atomistic systems containing 512 atoms.

Model	Acceleration	Time	Throughput
Prophet-OAME-MBD	openEquivariance	356.54 (337.61)	1.4k
TECE-OAM-RRA-1.0	OpenEquivariance, AOTInductor	124.92 (124.85)	4.1k
EquiformerV3+DeNS-OAM	-	144.18 (136.57)	3.6k
GRACE-3L-OAM-L	XLA JIT	77.62 (62.52)	6.6k
PET-OAM-XL	TorchScript	140.81 (118.38)	3.6k
TACE-OAM-L	OpenEquivariance, AOTInductor	41.09 (30.89)	12.5k
eSEN-30M-OAM	-	159.87 (150.83)	3.2k
NequIP-OAM-XL	OpenEquivariance, AOTInductor	13.59 (11.51)	37.7k
MatRIS-10M-OAM	-	327.24 (196.31)	1.6k
SevenNet-Omni-i12	cuEquivariance	99.50 (92.63)	5.1k
EquFlash-S-small	cuEquivariance, `torch.compile`	44.17 (33.38)	11.6k
EquFlash-S-medium	cuEquivariance, `torch.compile`	59.45 (49.07)	8.6k
EquFlash-S-large	cuEquivariance, `torch.compile`	92.56 (85.71)	5.5k

Table 14: Inference time (ms) and throughput (atoms/sec) of various models for atomistic systems containing 1024 atoms.

Model	Acceleration	Time	Throughput
Prophet-OAME-MBD	openEquivariance	434.22 (413.39)	2.4k
TECE-OAM-RRA-1.0	OpenEquivariance, AOTInductor	255.23 (235.91)	4.0k
EquiformerV3+DeNS-OAM	-	246.44 (235.53)	4.2k
GRACE-3L-OAM-L	XLA JIT	135.66 (116.66)	7.5k
PET-OAM-XL	TorchScript	253.15 (212.64)	4.0k
TACE-OAM-L	OpenEquivariance, AOTInductor	70.26 (54.82)	14.6k
eSEN-30M-OAM	-	241.08 (232.05)	4.3k
NequIP-OAM-XL	OpenEquivariance, AOTInductor	26.61 (21.58)	38.5k
MatRIS-10M-OAM	-	571.86 (280.55)	1.8k
SevenNet-Omni-i12	cuEquivariance	112.95 (109.37)	9.1k
EquFlash-S-small	cuEquivariance, `torch.compile`	43.05 (33.71)	23.8k
EquFlash-S-medium	cuEquivariance, `torch.compile`	63.97 (49.86)	16.0k
EquFlash-S-large	cuEquivariance, `torch.compile`	94.44 (85.83)	10.8k

Table 15: Inference time (ms) and throughput (atoms/sec) of various models for atomistic systems containing 2048 atoms.

Model	Acceleration	Time	Throughput
Prophet-OAME-MBD	openEquivariance	628.56 (626.90)	3.3k
TECE-OAM-RRA-1.0	OpenEquivariance, AOTInductor	501.43 (451.47)	4.1k
EquiformerV3+DeNS-OAM	-	OOM	
GRACE-3L-OAM-L	XLA JIT	261.01 (224.84)	7.8k
PET-OAM-XL	TorchScript	436.49 (402.50)	4.7k
TACE-OAM-L	OpenEquivariance, AOTInductor	124.40 (105.37)	16.5k
eSEN-30M-OAM	-	417.82 (400.49)	4.9k
NequIP-OAM-XL	OpenEquivariance, AOTInductor	56.92 (42.06)	36.0k
MatRIS-10M-OAM	-	972.05 (492.06)	2.1k
SevenNet-Omni-i12	cuEquivariance	117.75 (107.27)	17.4k
EquFlash-S-small	cuEquivariance, `torch.compile`	50.97 (33.93)	40.2k
EquFlash-S-medium	cuEquivariance, `torch.compile`	68.89 (52.48)	29.7k
EquFlash-S-large	cuEquivariance, `torch.compile`	105.59 (87.73)	19.4k

Table 16: Inference time (ms) and throughput (atoms/sec) of various models for atomistic systems containing 4096 atoms.

Model	Acceleration	Time	Throughput
Prophet-OAME-MBD	openEquivariance	OOM	
TECE-OAM-RRA-1.0	OpenEquivariance, AOTInductor	OOM	
EquiformerV3+DeNS-OAM	-	OOM	
GRACE-3L-OAM-L	XLA JIT	478.00 (413.40)	8.6k
PET-OAM-XL	TorchScript	OOM	
TACE-OAM-L	OpenEquivariance, AOTInductor	232.60 (206.34)	17.6k
eSEN-30M-OAM	-	OOM	
NequIP-OAM-XL	OpenEquivariance, AOTInductor	108.63 (80.55)	37.7k
MatRIS-10M-OAM	-	1816.19 (945.93)	2.3k
SevenNet-Omni-i12	cuEquivariance	129.26 (110.02)	31.7k
EquFlash-S-small	cuEquivariance, `torch.compile`	89.52 (34.73)	45.8k
EquFlash-S-medium	cuEquivariance, `torch.compile`	112.69 (57.12)	36.3k
EquFlash-S-large	cuEquivariance, `torch.compile`	174.71 (113.83)	23.4k

A.10 HYPERPARAMETERS: ARCHITECTURAL ANALYSIS

Table 17: Hyperparameter settings for the ablation studies. Unless otherwise specified, all configurations within each ablation use the same settings.

	Interaction Block (Table 1)	Message Passing (Table 2)	Node-wise FFN (Table 8)
Architecture			
Channels	128	128	128
$\ell_{\max}$	3	3†	3
Interaction blocks	5	5†	5
Cutoff radius (Å)	6.0	6.0	6.0
Radial basis	8	8	8
Training			
Max atoms	4000	4000	4000
Epochs		30	
Optimizer		AdamW	
LR scheduler		OneCycleLR(warm-up: 0.01)	
EMA decay		0.999	
Learning Rate			
Default	0.01	0.01 (CGTP)	0.01
Alternative	0.001 (Row 4)	0.005 (SO(2))	–

† Model scale is varied in the scaling experiments as specified in Table 2.

Unless otherwise specified, all ablation experiments follow the same training configuration described in Table 17. Hyperparameters are kept fixed within each ablation except where explicitly noted below. We use atom-based batching, where the batch size is determined by the total number of atoms rather than a fixed number of snapshots. Specifically, the "Max atoms" parameter specifies the maximum total number of atoms allowed in each batch, resulting in a variable number of snapshots per batch.

Interaction block ablation. For Table 1, all variants except the non-residual variant (Row 4) use the same training hyperparameters, including a learning rate of 0.01. The non-residual variant removes the normalization preceding the feed-forward network and uses a learning rate of 0.001. We found these adjustments necessary for stable optimization. When retaining normalization, loss reduction became extremely slow as model depth increased, even with larger learning rates. Removing normalization alleviated this issue, but larger learning rates caused deeper models to diverge. We therefore use a learning rate of 0.001 without normalization for the non-residual variant, while keeping all other training configurations unchanged.

Message-passing layer ablation. For Table 2, all CGTP-based models (Rows 1–6) use a learning rate of 0.01. For the SO(2)-based models, we selected the learning rate using the base configuration in Row 11. We evaluated learning rates of $0.01, 0.005, 0.002$ and found 0.005 to give the lowest validation loss. We therefore use 0.005 for all SO(2)-based configurations (Rows 7–11), while keeping the remaining training hyperparameters fixed.

Node-wise feed-forward ablation. For Table 8, all training and architectural hyperparameters other than the feed-forward component explicitly varied in the table are kept fixed. In particular, the same learning rate and optimization schedule are used for all nonlinearities and expansion ratios.

A.11 HYPERPARAMETERS: BENCHMARK EVALUATION

The hyperparameter settings used in our experiments are summarized in Tables 18 and 19. Table 18 provides the configurations for the MPTrj, OMat24, and MPTrj+sAlex training datasets across different model sizes, while Table 19 reports the configurations used for MatPES, OC20, and SPICE-MACE-OFF. Models reported as trained on the OAM dataset are, in most cases, pretrained on OMat24 and subsequently fine-tuned on MPTrj+sAlex. A dash (–) in the maximum number of neighbors column indicates that no limit is imposed. As in the ablation studies, "Max atoms" denotes the maximum total number of atoms per batch, rather than the number of snapshots. Configurations without this limit use a fixed batch size instead.

Table 18: EquFlash-S hyperparameter settings for MPTrj, OMat24, and MPTrj+sAlex.

	MPTrj			OMat24			MPTrj+sAlex		
	S	M	L	S	M	L	S	M	L
Architecture									
Parameters	4.5M	10.3M	15.8M	4.5M	10.3M	15.5M	4.5M	10.3M	15.5M
Channels	128	128	128	128	128	128	128	128	128
$\ell_{\max}$	3	3	4	3	3	4	3	3	4
Layers	5	10	10	5	10	10	5	10	10
Cutoff radius (Å)	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0
Radial basis	8	8	8	8	8	8	8	8	8
Training									
Max atoms	8000	8000	8000	16000	16000	16000	16000	16000	16000
Epochs	100	100	30†+70	6	6	6	1	1	1
Energy weight		1			1			1	
Force weight		1			1			1	
Shared Hyperparameters									
Learning rate	0.01								
LR Scheduler	OneCycleLR (warm-up: 0.01)								
Optimizer	AdamW								
EMA decay	0.999								

† denotes direct force pretraining.

Table 19: EquFlash-S hyperparameter settings for MatPES, OC20, and SPICE.

	MatPES	OC20	SPICE-MACE-OFF	
Architecture				
Parameters	1.4M	228M	3.0M	6.6M
Channels	64	320	128	128
$\ell_{\max}$	3	5	2	2
Layers	5	16	5	10
Cutoff radius (Å)	6.0	12.0	6.0	6.0
Maximum number of neighbors	-	20	-	-
Radial basis	8	64	8	8
Training				
Max atoms	–	1200	–	–
Batch size	64	–	128	128
Epochs	30	12	100	100
Energy weight	1	1	1	1
Force weight	1	1000	1	1
Shared Hyperparameters				
Learning rate	0.01			
LR Scheduler	OneCycleLR (warm-up: 0.01)			
Optimizer	AdamW			
EMA decay	0.999			