MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields

Ilyes Batatia
Engineering Laboratory,
University of Cambridge
Cambridge, CB2 1PZ UK
Department of Chemistry,
ENS Paris-Saclay, Université Paris-Saclay
91190 Gif-sur-Yvette, France
ilyes.batatia@ens-paris-saclay.fr

Dávid Péter Kovács
Engineering Laboratory,
University of Cambridge
Cambridge, CB2 1PZ UK

Gregor N. C. Simm
Engineering Laboratory,
University of Cambridge
Cambridge, CB2 1PZ UK

Christoph Ortner
Department of Mathematics
University of British Columbia
Vancouver, BC, Canada V6T 1Z2

Gábor Csányi
Engineering Laboratory,
University of Cambridge
Cambridge, CB2 1PZ UK

Abstract

Creating fast and accurate force fields is a long-standing challenge in computational chemistry and materials science. Recently, several equivariant message passing neural networks (MPNNs) have been shown to outperform models built using other approaches in terms of accuracy. However, most MPNNs suffer from high computational cost and poor scalability. We propose that these limitations arise because MPNNs only pass two-body messages leading to a direct relationship between the number of layers and the expressivity of the network. In this work, we introduce MACE, a new equivariant MPNN model that uses higher body order messages. In particular, we show that using four-body messages reduces the required number of message passing iterations to just *two*, resulting in a fast and highly parallelizable model, reaching or exceeding state-of-the-art accuracy on the rMD17, 3BPA, and AcAc benchmark tasks. We also demonstrate that using higher order messages leads to an improved steepness of the learning curves.

1 Introduction

The earliest approaches for creating force fields (interatomic potentials) using machine learning techniques were using local atom-centered symmetric descriptors and feed-forward neural networks [6], Gaussian Process regression[2] or linear regression [44, 47]. The first attempts to use graph neural networks to model the potential energy of atomistic systems had only limited success. The DTNN [42], SchNet [41], HIP-NN [35], PhysNet [48], or DimeNet [20, 29] approaches could only come close to but not improve upon the atomic descriptor-based methods in terms of computational efficiency

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

and accuracy on public benchmarks. Furthermore, most MPNN interatomic potentials use 2-body invariant messages, making them non-universal approximators [38].

The MACE architecture presented here allows for the efficient computation of equivariant messages with high body order. As a result of the increased body order of the messages, only two message passing iterations are necessary to achieve high accuracy - unlike the typical five or six iterations of MPNNs, making it scalable and parallelizable. Finally, our implementation has remarkable computational efficiency, reaching state-of-the-art results on the 3BPA benchmark after 30 mins of training on NVIDIA A100 GPUs.

We summarise our main contributions as follows:

- We introduce MACE, a novel architecture combining equivariant message passing with efficient many-body messages. The MACE models achieve state-of-the-art performance on challenging benchmark tests. They also display greater generalization capabilities over other approaches on extrapolation benchmarks.
- We demonstrate that many-body messages change the power of the empirical power-law of the learning curves. Furthermore, we show experimentally that the addition of equivariant messages only shifts the learning curves but does not change the power law when higher order messages are used.
- We show that MACE does not only outperform previous approaches in terms of accuracy but also does so while being significantly faster to train and evaluate than the previous most accurate models.

2 Background

2.1 MPNN Interatomic Potentials

MPNNs [22, 9] are a type of graph neural network (GNN, [40, 4, 27, 51]) that parametrises a mapping from a labeled graph to a target space, either a graph or a vector space. When applied to parameterise properties of atomistic structures (materials or molecules), the graph is embedded in 3-dimensional (3D) Euclidean space, where each node represents an atom, and edges connect nodes if the corresponding atoms are within a given distance of each other. We represent the state of each node i in layer t of the MPNN by a tuple

$$\sigma_i^{(t)} = (\boldsymbol{r}_i, z_i, \boldsymbol{h}_i^{(t)}), \tag{1}$$

where $\boldsymbol{r}_i \in \mathbb{R}^3$ is the position of atom i, z_i the chemical element, and $\boldsymbol{h}_i^{(t)}$ are its learnable features. A forward pass of the network consists of multiple *message construction*, *update*, and *readout* steps. During message construction, a message $\boldsymbol{m}_i^{(t)}$ is created for each node by pooling over its neighbors:

$$\boldsymbol{m}_i^{(t)} = \bigoplus_{j \in \mathcal{N}(i)} M_t(\sigma_i^{(t)}, \sigma_j^{(t)}), \tag{2}$$

where M_t is a learnable message function and $\bigoplus_{j \in \mathcal{N}(i)}$ is a learnable, permutation invariant pooling operation over the neighbors of atom i (e.g., a sum). In the update step, the message $\boldsymbol{m}_i^{(t)}$ is transformed into new features

$$\boldsymbol{h}_i^{(t+1)} = U_t(\sigma_i^{(t)}, \boldsymbol{m}_i^{(t)}), \tag{3}$$

where U_t is a learnable update function. After T message construction and update steps, the learnable readout functions $\mathcal{R}_t$ map the node states $\sigma_i^{(t)}$ to the target, in this case the site energy of atom i,

$$E_i = \sum_{t=1}^{T} \mathcal{R}_t(\sigma_i^{(t)}). \tag{4}$$

2.2 Equivariant Graph Neural Networks

In *equivariant* GNNs, internal features $\boldsymbol{h}_i^{(t)}$ transform in a specified way under some group action [1, 12, 32, 46, 49]. When modelling the potential energy of an atomic structure, the group of interest is

O(3), specifying rotations and reflections of the particles.[1] We call a GNN O(3) equivariant if it has internal features that transform under the rotation $Q \in \mathrm{O}(3)$ as

$$\boldsymbol{h}_i^{(t)}(Q \cdot (\boldsymbol{r}_1, ..., \boldsymbol{r}_N)) = D(Q)\boldsymbol{h}_i^{(t)}(\boldsymbol{r}_1, ..., \boldsymbol{r}_N), \tag{5}$$

where $Q \cdot (\boldsymbol{r}_1, ..., \boldsymbol{r}_N)$ denotes the action of the rotation on the set of atomic positions and $D(Q)$ is a matrix representing the rotation Q, acting on message $\boldsymbol{h}_i^{(t)}$. In general, elements of the feature vector can be labeled according to the irreducible representation they transform with. We will write $h_{i,kLM}^{(t)}$ to indicate a collection of features on atom i, indexed by k, that transform according to

$$h_{i,kLM}^{(t)}(Q \cdot (\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N)) = \sum_{M'} D_{M'M}^{L}(Q) h_{i,kLM'}^{(t)}(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N), \tag{6}$$

where $D^L(Q) \in \mathbb{R}^{(2L+1)\times(2L+1)}$ is a Wigner D-matrix of order L. A feature labelled with $L = 0$ describes an invariant scalar. Features labeled with $L > 0$, describe equivariant features, formally corresponding to equivariant vectors, matrices or higher order tensors. The features of *invariant* models, such as SchNet[41] and DimeNet[29], transform according to $D(Q) = \mathbb{1}$, the identity matrix. Models such as NequIP [5], equivariant transformer [45], PaiNN [43], or SEGNNs [8], in addition to invariant scalars, employ equivariant internal features that transform like vectors or tensors.

3 Related Work

ACE - Higher Order Local Descriptors In the last few years, there have been two significant breakthroughs in machine learning force fields. First, the Atomic Cluster Expansion (ACE) [16] provided a systematic framework for constructing high body order complete polynomial basis functions (features) at a constant cost per basis function, independent of body order [17]. It has also been shown that ACE includes many previously developed atomic environment representations as special cases, including Atom Centred Symmetry Functions [6], the Smooth Overlap of Atomic Positions (SOAP) descriptor [2], Moment Tensor Potential basis functions [44], and the hyperspherical bispectrum descriptor [2] used by the SNAP model [47]. These local models are limited by their cutoff distance and their relatively rigid architecture compared to the overparametrised MPNNs, leading to somewhat lower accuracy, in particular, for molecular force fields.

Equivariant MPNNs The second breakthrough was using equivariant internal features in MPNNs. These equivariant MPNNs, such as Cormorant [1], Tensor Field Networks [46], EGNN [39], PaiNN [43], Equivariant Transformers [45], SEGNN [8], NewtonNet [23], and NequIP [5] were able to achieve higher performance than previous local descriptor-based models. However, they suffer from two significant limitations: first, the most accurate models used $L = 3$ spherical tensors as messages and 4 to 6 message passing iterations [5], which resulted in a relatively high computational cost. Second, using this many iterations significantly increased the receptive field of the network, making them difficult to parallelise across multiple GPUs [36].

Higher Order Message Passing Most MPNNs use a message passing scheme based on two-body messages, meaning they simultaneously depend on the states of two atoms. It has been recognised that it can be beneficial to include angles into the features, effectively creating 3-body invariant messages [29]. This idea has also been exploited in other invariant MPNNs, in particular, by SphereNet [34] and GemNet [30]. Even though these models improved the accuracy compared to the 2-body message passing, they were limited by the computational cost associated with explicitly summing over all triplets or quadruplets to compute the higher order features.

Multi-ACE Framework Recently, multi-ACE has been proposed as a unifying framework of $E(3)$-equivariant atom-centered interatomic potentials, extending the ACE framework to include methods built on equivariant MPNNs [3]. A similar unifying theories were also put forward by [37] and [7]. The idea is to parameterise the message $\boldsymbol{m}_i^{(t)}$ in terms of invariant or equivariant ACE models. This framework sets out a design space in which each model can be characterised in terms of: (1) the number of layers, (2) the body order of the messages, (3) the equivariance (or invariance) of the messages, and (4) the number of features in each layer. The framework highlights the relationship between the overall body order of the models and message passing, also previously

[1] Translation invariance is trivially incorporated through the use of relative distances.

discussed in Kondor [31]. Most previously published models achieved high accuracy by *either* using 4 to 6 layers [5, 43] *or* increasing the local body order with a single layer [33, 36]. With our model, we fall in between these two extremes by combining high body order with message passing.

4 The MACE Architecture

Our MACE model follows the general framework of MPNNs outlined in Section 2. Our key innovation is a new message construction mechanism. We expand the messages $\boldsymbol{m}_i^{(t)}$ in a hierarchical body order expansion,

$$\boldsymbol{m}_i^{(t)} = \sum_j \boldsymbol{u}_1\left(\sigma_i^{(t)}; \sigma_j^{(t)}\right) + \sum_{j_1,j_2} \boldsymbol{u}_2\left(\sigma_i^{(t)}; \sigma_{j_1}^{(t)}, \sigma_{j_2}^{(t)}\right) + \cdots + \sum_{j_1,\ldots,j_\nu} \boldsymbol{u}_\nu\left(\sigma_i^{(t)}; \sigma_{j_1}^{(t)}, \ldots, \sigma_{j_\nu}^{(t)}\right), \tag{7}$$

where the $\boldsymbol{u}$ functions are learnable, the sums run over the neighbors of i, and ν is a hyper-parameter corresponding to the maximum correlation order, the body order minus 1, of the message function with respect to the states. Even though we refer to the message as $(\nu+1)$-body with respect to the states, the overall body order with respect to the positions can be larger depending on the body order of the states themselves. Crucially, by writing $\sum_{j_1,\ldots,j_\nu}$, which includes self-interaction (e.g., $j_1 = j_2$), we will later obtain a tensor product structure with a computationally efficient parameterisation, that allows us to circumvent the seemingly exponential scaling of the computational cost with the correlation order ν. This contrasts with previous models, such as DimeNet [28, 29], that compute 3-body features via the more standard many-body expansion $\sum_{j_1<\cdots<j_\nu}$. Below, we describe the MACE architecture in detail. To better understand the architecture, we report in A.4 a table of the introduced tensors along with their shapes.

Message Construction At each iteration, we first embed the edges using a learnable radial basis $R_{kl_1l_2l_3}^{(t)}$, a set of spherical harmonics $Y_{l_1}^{m_1}$, and a learnable embedding of the previous node features $h_{j,\tilde{k}l_2m_2}^{(t)}$ using weights $W_{k\tilde{k}l_2}^{(t)}$. The $\boldsymbol{A}_i^{(t)}$-features are obtained by pooling over the neighbours $\mathcal{N}(i)$ to obtain permutation invariant 2-body features whilst, crucially, retaining full directional information, and thus, full information about the atomic environment:

$$A_{i,kl_3m_3}^{(t)} = \sum_{l_1m_1,l_2m_2} C_{l_1m_1,l_2m_2}^{l_3m_3} \sum_{j\in\mathcal{N}(i)} R_{kl_1l_2l_3}^{(t)}(r_{ji}) Y_{l_1}^{m_1}(\hat{\boldsymbol{r}}_{ji}) \sum_{\tilde{k}} W_{k\tilde{k}l_2}^{(t)} h_{j,\tilde{k}l_2m_2}^{(t)}, \tag{8}$$

where $C_{l_1m_1,l_2m_2}^{l_3m_3}$ are the standard Clebsch-Gordan coefficients ensuring that $A_{i,kl_3m_3}^{(t)}$ maintain the correct equivariance, r_{ji} is the (scalar) interatomic distance, and $\hat{\boldsymbol{r}}_{ji}$ is the corresponding unit vector. $R_{kl_1l_2l_3}^{(t)}$ is obtained by feeding a set of radial features that embed the radial distance r_{ji} using Bessel functions multiplied by a smooth polynomial cutoff (cf. Ref. [29]) to a multi-layer perceptron (MLP). See Section A.5 for details. In the first layer, the node features $h_j^{(t)}$ correspond to the (invariant) chemical element z_j. Therefore, (8) can be further simplified:

$$A_{i,kl_1m_1}^{(1)} = \sum_{j\in\mathcal{N}(i)} R_{kl_1}^{(1)}(r_{ji}) Y_{l_1}^{m_1}(\hat{\boldsymbol{r}}_{ji}) W_{kz_j}^{(1)}. \tag{9}$$

This simplified operation is much cheaper, making the computational cost of the first layer low.

The *key* operation of MACE is the efficient construction of higher order features from the $\boldsymbol{A}_i^{(t)}$-features. This is achieved by first forming tensor products of the features, and then symmetrising:

$$\boldsymbol{B}_{i,\eta_\nu kLM}^{(t)} = \sum_{\boldsymbol{lm}} \mathcal{C}_{\eta_\nu,\boldsymbol{lm}}^{LM} \prod_{\xi=1}^{\nu} \sum_{\tilde{k}} w_{k\tilde{k}l_\xi}^{(t)} A_{i,\tilde{k}l_\xi m_\xi}^{(t)}, \quad \boldsymbol{lm} = (l_1m_1, \ldots, l_\nu m_\nu) \tag{10}$$

where the coupling coefficients $\mathcal{C}_{\eta_\nu}^{LM}$ corresponding to the generalised Clebsch-Gordan coefficients (details in A.3) ensuring that $\boldsymbol{B}_{i,\eta_\nu kLM}^{(t)}$ are L-equivariant, the weights $w_{k\tilde{k}l_\xi}^{(t)}$ are mixing the channels (k) of $\boldsymbol{A}_i^{(t)}$, and ν is a given correlation order. $\mathcal{C}_{\eta_\nu,\boldsymbol{lm}}^{LM}$ is very sparse and can be pre-computed such that (10) can be evaluated efficiently (see Appendix A.3.3). The additional index η_ν simply enumerates all possible couplings of $l_1, \ldots, l_\nu$ features that yield the selected equivariance specified

by the L index. The $\boldsymbol{B}_i^{(t)}$-features are constructed up to some maximum ν. This variable in (10) is the order of the tensor product, and hence, can be identified as the order of the many-body expansion terms in (7). The computationally expensive multi-dimensional sums over all triplets, quadruplets, etc., are thus circumvented and absorbed into (9) and (8).

The message $\boldsymbol{m}_i^{(t)}$ can now be written as a linear expansion

$$m_{i,kLM}^{(t)} = \sum_{\nu} \sum_{\eta_\nu} W_{z_i kL,\eta_\nu}^{(t)} \boldsymbol{B}_{i,\eta_\nu kLM}^{(t)}, \tag{11}$$

where $W_{z_i kL,\eta_\nu}^{(t)}$ is a learnable weight matrix that depends on the chemical element z_i of the receiving atom and message symmetry L. Thus, we implicitly construct each term $\boldsymbol{u}$ in (7) by a linear combination of $\boldsymbol{B}_{i,\eta_\nu kLM}^{(t)}$ features of the corresponding body order.

Under mild conditions on the two-body bases $\boldsymbol{A}_i^{(t)}$, the higher order features $\boldsymbol{B}_{i,\eta_\nu kLM}^{(t)}$ can be interpreted as a *complete basis* of many-body interactions [17], which can be computed at a cost comparable to pairwise interactions. Because of this, the expansion (11) is *systematic*. It can in principle be converged to represent any smooth $(\nu+1)$-body equivariant mapping in the limit of infinitely many features (proof in [17]).

Update In MACE, the update is a linear function of the message and the residual connection [25]:

$$h_{i,kLM}^{(t+1)} = U_t^{kL}(\sigma_i^{(t)}, \boldsymbol{m}_i^{(t)}) = \sum_{\tilde{k}} W_{kL,\tilde{k}}^{(t)} m_{i,\tilde{k}LM}^{(t)} + \sum_{\tilde{k}} W_{z_i kL,\tilde{k}}^{(t)} h_{i,\tilde{k}LM}^{(t)}. \tag{12}$$

Readout In the readout phase, the invariant part of the node features is mapped to a hierarchical decomposition of site energies via readout functions:

$$\begin{aligned} E_i &= E_i^{(0)} + E_i^{(1)} + ... + E_i^{(T)}, \qquad \text{where} \\ E_i^{(t)} &= \mathcal{R}_t\left(\boldsymbol{h}_i^{(t)}\right) = \begin{cases} \sum_{\tilde{k}} W_{\text{readout},\tilde{k}}^{(t)} h_{i,\tilde{k}00}^{(t)} & \text{if } t < T \\ \text{MLP}_{\text{readout}}^{(t)}\left(\left\{h_{i,k00}^{(t)}\right\}_k\right) & \text{if } t = T \end{cases} \end{aligned} \tag{13}$$

The readout only depends on the invariant features $h_{i,k00}^{(t)}$ to ensure that the site energy contributions $E_i^{(t)}$ are invariant as well. To maintain body ordering, we use linear readout functions for all layers except the last, where we use a one-layer MLP.

5 Results

5.1 Effect of Higher Order Messages

Number of Layers In this section, we investigate the effect of using higher order messages. Many MPNN architectures [41, 48] exclusively pass two-body invariant messages resulting in an incomplete representation of the local environment [38]. Equivariant message-passing schemes [5, 43, 8] lift the degeneracy of most structures by containing directional information in the messages. MPNNs that only employ two-body messages at each layer can increase the body order *either* by stacking layers [31] which simultaneously increases the model's receptive field *or* by using non-linear activation functions, generate only a subset of all possible higher order features. By constructing higher order messages using the MACE architecture, we disentangle the increase in body order from the increase of the receptive field.

In Figure 1, we show the accuracy of MACE, NequIP, and BOTNet [3] on the 3BPA benchmark [33] as a function of the number of message passing layers. Approaches employing 2-body message passing require up to five iterations for their accuracy to converge. By constructing many body messages, the number of required layers to converge in accuracy reduces to just two. In all subsequent experiments, we use two-layer MACE models.

Furthermore, we compare BOTNet, which does not use any non-linearities in the update step to NequIP, which does. Otherwise, the two models are very similar. We observe that the increase in body

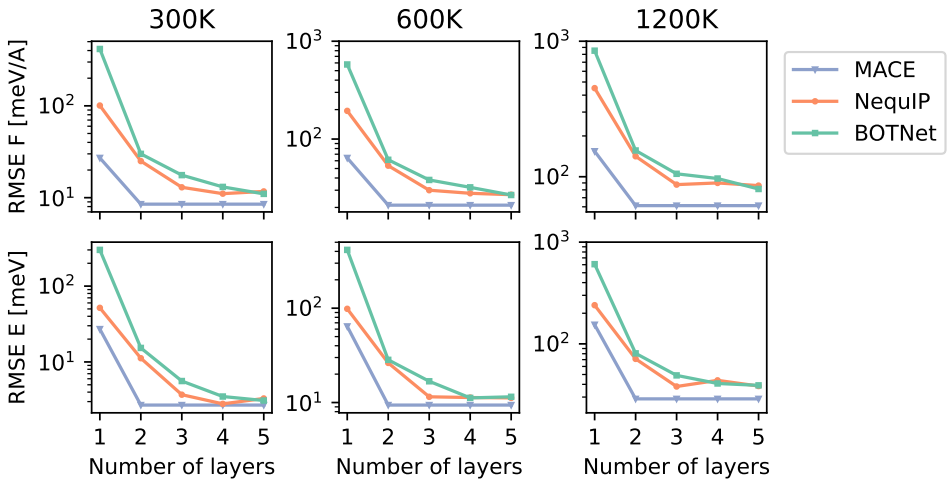

Figure 1: Energy and force errors of BOTNet, NequIP, and MACE ($L = 2$) on the 3BPA dataset at different temperatures as a function of the number of layers.

order through non-linearities within the update provides only marginal improvement, highlighting the difference between an increase in body order through non-linearities (NequIP) and higher order symmetric messages (MACE). Consequently, higher order message passing allows one to reduce the number of layers, thereby increasing speed and ease of parallelization over multiple GPUs. We note that MACE does not improve after two layers as the diameter of the 3BPA molecule is about 9 Å and radial cutoff in each layer is 5 Å.

Learning Curves We study how higher order message passing affects the learning curves. A recent study of the NequIP model [5] showed that the inclusion of equivariant features results in enhanced data efficiency, increasing the slope of the log-log plot of predictive error as a function of the dataset size. They showed that adding equivariance not only *shifts* the learning curves, but also changes the powers in the empirical power law of the learning curves, which is usually constant for a given dataset [26].

On the left panel of Figure 2, we replicate the experiments of [5] by training a series of *invariant* MACE models with increasing correlation order ν on the aspirin molecule from the rMD17 dataset. We observe that adding higher order messages changes the steepness of the learning curves, even without equivariant messages. The model with correlation order $\nu = 1$ corresponds to a two-layer 2-body invariant model, similar to SchNet. This model is the least accurate due to the incomplete nature of 2-body invariant representations of the local environment [38]. The invariant messages with $\nu = 2$ are akin to those in DimeNet, which explicitly puts angular information into the messages. We see that including higher order information significantly improves the model's accuracy. Finally, by going beyond any current message passing potential by setting $\nu = 3$, we achieve similar performance to a highly-accurate 2-body, equivariant MPNN while only using higher order invariant messages.

On the middle panel of Figure 2, we keep the correlation order fixed at $\nu = 3$ and gradually increase the symmetry order L of the messages. While the slope remains nearly unchanged, the curves are shifted. In the right panel of Figure 2, we keep the correlation order fixed at $\nu = 1$ and gradually increase the symmetry order L of the messages. We see only a marginal slope change when adding equivariant features, which could be attributable to the relatively low expressiveness of a two-layer MACE restricted to correlation order $\nu = 1$. These results suggest two routes to improve invariant 2-body MPNN models: creating higher correlation order messages or incorporating equivariant messages. By exploiting both of these options, the MACE model achieves state-of-the-art accuracy.

5.2 Scaling and Computational Cost

Chemical Elements A significant limitation of existing atomic environment representations is that their size grows with the number of chemical elements S and correlation order ν as S^ν. Data-driven compression schemes have been proposed [50] to solve this issue, and MPNNs incorporate similar

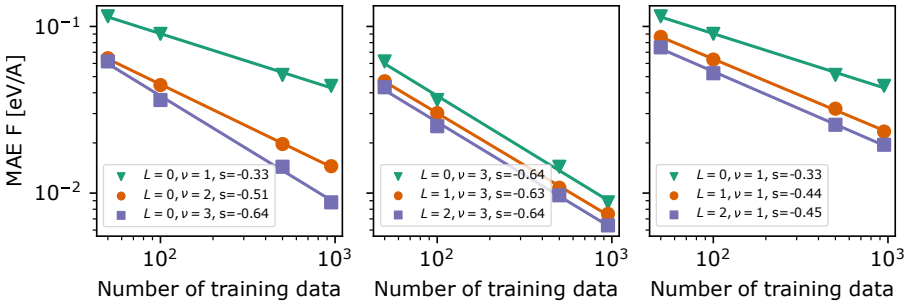

Figure 2: Learning curve of force errors (MAE in eV / Å) for aspirin from the rMD17 dataset for different models. *Left:* Two layers of invariant ($L = 0$) MACE with increasing body order $\nu \in \{1, 2, 3\}$. *Center:* Two layers of MACE with $\nu = 3$ and increasing equivariance $L \in \{0, 1, 2\}$. *Right:* Two layers of MACE with $\nu = 1$ and increasing equivariance $L \in \{0, 1, 2\}$. In each case the slope (s) is indicated.

embeddings of the chemical elements into a fixed-size vector space. MACE uses a continuous species embedding and when constructing the higher order features in (10), it does not include the species dimension k in the tensor product resulting in $\mathcal{O}(1)$ scaling of the model with the number of chemical elements S.

Receptive Field A severe limitation of many previously published MPNNs was their large receptive field, making it difficult to parallelize the evaluation across multiple GPUs. In traditional MPNNs, the total receptive field of each node, which grows with each message passing iteration, can be up to 30 Å. This scaling results in the number of neighbours being in the thousands in a condensed phase simulation, preventing any efficient parallelization [36]. By decoupling the increase in correlation order of the messages from the number of message passing iterations, MACE only requires two layers resulting in a much smaller receptive field. With a local radial cutoff of 4 to 5 Å, the overall receptive field remains small, making the model more parallelisable.

Computational Cost The computational bottleneck of equivariant MPNNs is the equivariant tensor product (8). This tensor product is evaluated on edges. In MACE, we only evaluate this expensive tensor product once, within the second layer, and build up correlations through the tensor product of (10). Importantly, this operation is carried out on nodes. Typically the number of nodes is orders of magnitudes smaller than the number of edges resulting in a computational advantage. In addition, we developed a loop tensor contraction algorithm for the efficient implementation of (10) and (11) detailed in Section A.3.

We report evaluation times for BOTNet, NequIP, and multiple versions of MACE in Table 2. We observe that the invariant MACE ($L = 0$) is close to 10 times faster than BOTNet and NequIP while achieving similar accuracy at high temperatures. MACE with $L = 1$ and $L = 2$ is 5 and 4 times faster than BOTNet and NequIP, respectively, while outperforming them at every temperature. We acknowledge that accurate speed comparisons between codes are hard to obtain, and further investigations need to be carried out. It is also essential to consider training times. In order to do a fair comparison, all the timings were realised using the `mace` code that implements all the above models. Models that are significantly faster to train are better suited for applications of active learning, which is typically how databases for materials science applications are built [13–15]. The MACE model reported in Table 2 takes approximately 30 mins to reach the accuracy of a converged BOTNet model, taking more than a day to be trained on the 3BPA dataset using NVIDIA A100 GPUs.

5.3 Benchmark Results [2]

5.3.1 rMD17: Molecular Dynamics Trajectory

The revised MD17 (rMD17) dataset contains train test splits randomly selected from a long molecular dynamics trajectory of ten small organic molecules [11]. For each molecule, the splits consist of 1000 training and test configurations. Table 1 shows that MACE achieves excellent accuracy, improving

[2]Training details and hyper-parameters for all experiments can be found in Appendix A.5

Table 1: **Mean absolute errors on the rMD17 dataset** [11]. Energy (E, meV) and force (F, meV/Å) errors of different models trained on 950 configurations and validated on 50. The models on the right of the first vertical line, DimeNet and NewtonNet, were trained on the original MD17 dataset [10]. The models on the right of the second (double) vertical line were trained on just 50 configurations.

		MACE	Allegro [36]	BOTNet [3]	NequIP [5]	GemNet (T/Q) [30]	ACE [33]	FCHL [18]	GAP [2]	ANI [19]	PaiNN [43]	DimeNet [29]	NewtonNet [24]	ACE [33]	NequIP [5]	MACE
		$N_{\text{train}} = 1000$												$N_{\text{train}} = 50$		
Aspirin	E	**2.2**	2.3	2.3	2.3	-	6.1	6.2	17.7	16.6	6.9	8.8	7.3	26.2	19.5	**17.0**
	F	**6.6**	7.3	8.5	8.2	9.5	17.9	20.9	44.9	40.6	16.1	21.6	15.1	63.8	52.0	**43.9**
Azobenzene	E	1.2	1.2	**0.7**	**0.7**	-	3.6	2.8	8.5	15.9	-	-	6.1	9.0	6.0	**5.4**
	F	3.0	**2.6**	3.3	2.9	-	10.9	10.8	24.5	35.4	-	-	5.9	28.8	20.0	**17.7**
Benzene	E	0.4	0.3	**0.03**	0.04	-	0.04	0.35	0.75	3.3	-	3.4	-	**0.2**	0.6	0.7
	F	0.3	**0.2**	0.3	0.3	0.5	0.5	2.6	6.0	10.0	-	8.1	-	**2.7**	2.9	**2.7**
Ethanol	E	**0.4**	**0.4**	**0.4**	**0.4**	-	1.2	0.9	3.5	2.5	2.7	2.8	2.6	8.6	8.7	**6.7**
	F	**2.1**	**2.1**	3.2	2.8	3.6	7.3	6.2	18.1	13.4	10.0	10.0	9.1	43.0	40.2	**32.6**
Malonaldehyde	E	0.8	**0.6**	0.8	0.8	-	1.7	1.5	4.8	4.6	3.9	4.5	4.1	12.8	12.7	**10.0**
	F	4.1	**3.6**	5.8	5.1	6.6	11.1	10.3	26.4	24.5	13.8	16.6	14.0	63.5	52.5	**43.3**
Naphthalene	E	0.5	**0.2**	**0.2**	0.9	-	0.9	1.2	3.8	11.3	5.1	5.3	5.2	3.8	**2.1**	**2.1**
	F	1.6	**0.9**	1.8	1.3	1.9	5.1	6.5	16.5	29.2	3.6	9.3	3.6	19.7	10.0	**9.2**
Paracetamol	E	**1.3**	1.5	**1.3**	1.4	-	4.0	2.9	8.5	11.5	-	-	6.1	13.6	14.3	**9.7**
	F	**4.8**	4.9	5.8	5.9	-	12.7	12.3	28.9	30.4	-	-	11.4	45.7	39.7	**31.5**
Salicylic acid	E	0.9	0.9	0.8	**0.7**	-	1.8	1.8	5.6	9.2	4.9	5.8	4.9	8.9	8.0	**6.5**
	F	3.1	**2.9**	4.3	4.0	5.3	9.3	9.5	24.7	29.7	9.1	16.2	8.5	41.7	35.0	**28.4**
Toluene	E	0.5	0.4	**0.3**	**0.3**	-	1.1	1.7	4.0	7.7	4.2	4.4	4.1	5.3	3.3	**3.1**
	F	**1.5**	1.8	1.9	1.6	2.2	6.5	8.8	17.8	24.3	4.4	9.4	3.8	27.1	15.1	**12.1**
Uracil	E	0.5	0.6	**0.4**	**0.4**	-	1.1	0.6	3.0	5.1	4.5	5.0	4.6	6.5	7.3	**4.4**
	F	2.1	**1.8**	3.2	3.1	3.8	6.6	4.2	17.6	21.4	6.1	13.1	6.4	36.2	40.1	**25.9**

the state of the art for some molecules, particularly those with the highest errors. As several methods achieve similar accuracy on the standard task of predicting energies and forces based on the whole training set, we also trained MACE and NequIP, another accurate model, on just 50 configurations to increase the difficulty of the benchmark. In this case, we found that MACE outperformed NequIP for most molecules.

5.3.2 3BPA: Extrapolation to Out-of-domain Data

The 3BPA dataset introduced in [33] tests a model's extrapolation capabilities. Its training set contains 500 geometries sampled from 300 K molecular dynamics simulation of the large and flexible drug-like molecule 3-(benzyloxy)pyridin-2-amine. The three test sets contain geometries sampled at 300 K, 600 K, and 1200 K to assess in- and out-of-domain accuracy. A fourth test set consists of optimized geometries, where two of the molecule's dihedral angles are fixed, and a third is varied between 0 and 360 degrees resulting in so-called *dihedral slices* through regions of the PES far away from the training data.

The root-mean-squared errors (RMSE) on energies and forces for several models are shown in Table 2. It can be seen that MACE outperforms the other models on all tasks. In particular, when extrapolating to 1200 K data, MACE with $L = 2$ outperforms NequIP and Allegro models by about 30%. Further, MACE with $L = 2$ outperforms the next best model, BOTNet, by 40% on energies for the dihedral slices. Finally, the MACE model with invariant messages ($L = 0$) often nearly matches or exceeds the performance of competitive equivariant models.

Table 2: **Root-mean-square errors on the 3BPA dataset.** Energy (E, meV) and force (F, meV/Å) errors of models trained and tested on configurations collected at 300 K of the flexible drug-like molecule 3-(benzyloxy)pyridin-2-amine (3BPA). Standard deviations are computed over three runs and shown in brackets if available. In order to facilitate measuring the efficiency of *architectures* we implemented the NequIP and BOTNet architectures in the same code that we used for MACE and which is published together with this paper. For the precise specification of our NequIP implementation see the Appendix A.5.2. All PyTorch timings were realised on an NVIDIA A100 GPU custom implementations.

		Allegro (L=3)	**NequIP (L=3)**	**NequIP (L=3)**	**BOTNet (L=3)**	**MACE (L=0)**	**MACE (L=1)**	**MACE (L=2)**
Code		allegro [36]	nequip [5]	mace	mace	mace	mace	mace
300 K	E	3.84 (0.08)	3.3 (0.1)	3.1 (0.1)	3.1 (0.13)	4.5 (0.25)	3.4 (0.2)	**3.0** (0.2)
	F	12.98 (0.17)	10.8 (0.2)	11.3 (0.2)	11.0 (0.14)	14.6 (0.5)	10.3 (0.3)	**8.8** (0.3)
600 K	E	12.07 (0.45)	11.2 (0.1)	11.3 (0.31)	11.5 (0.6)	13.7 (0.16)	9.9 (0.8)	**9.7** (0.5)
	F	29.17 (0.22)	26.4 (0.1)	27.3 (0.3)	26.7 (0.29)	33.3 (1.35)	24.6 (1.1)	**21.8** (0.6)
1200 K	E	42.57 (1.46)	38.5 (1.6)	40.8 (1.3)	39.1 (1.1)	37.1 (0.8)	31.7 (0.5)	**29.8** (1.0)
	F	82.96 (1.77)	76.2 (1.1)	86.4 (1.5)	81.1 (1.5)	81.6 (3.89)	67.8 (1.8)	**62.0** (0.7)
Dihedral Slices	E	-	-	23.2	16.3 (1.5)	12.3 (0.8)	11.5 (0.6)	**7.8** (0.6)
	F	-	-	23.1	20.0 (1.2)	26.1 (2.8)	19.3 (0.6)	**16.5** (1.7)
Time latency [ms]		-	-	103.5	101.2	**10.5**	17.5	24.3

MACE shows excellent results while also featuring low computational cost compared to many other models. The $L = 0$ model, which approaches previous models in terms of accuracy, outpaces them by nearly a factor of 10, whereas the $L = 2$ model achieves state-of-the-art accuracy and is around four times faster than other equivariant MPNN models. In the table, we characterise the evaluation speed of the models by reporting the "latency" which is defined as the time it takes to compute forces on a structure, which is typically independent of the number of atoms until GPU threads are filled (typically 10,000 atoms for these models on an Nvidia A100 80GB GPU).

In Figure 3, we compare the BOTNet, NequIP, and MACE ($L = 2$) by inspecting their energy profile for three dihedral slices. Overall, it can be seen that all models produce smooth energy profiles and that, in general, MACE comes closest to the ground truth. The fact that MACE outperforms the

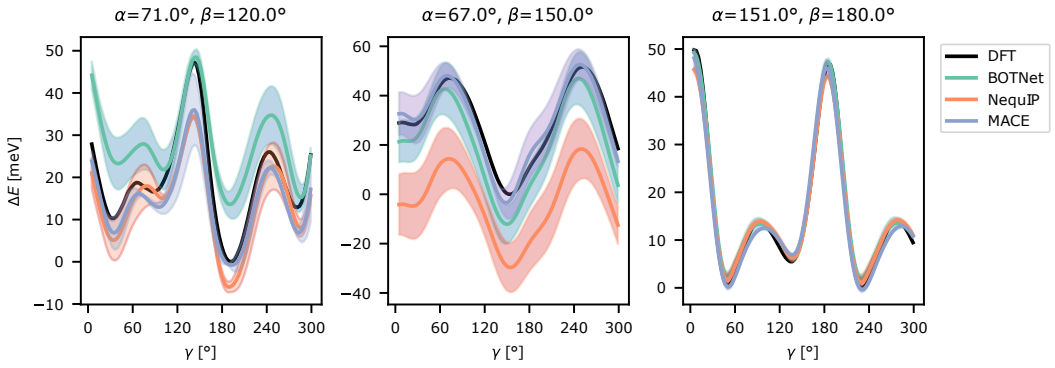

Figure 3: Energy predictions on three cuts through the potential energy surface of the 3-(benzyloxy)pyridin-2-amine (3BPA) molecule by BOTNet, NequIP, and MACE ($L = 2$). The ground-truth energy (DFT) is shown in black. For each cut, the curves have been shifted vertically so that the lowest ground-truth energy is zero.

other methods in the middle panel, which contains geometries furthest from the training dataset [3], suggests superior extrapolation capabilities.

5.3.3 AcAc: Flexibility and Reactivity

A similar benchmark dataset assessing a model's extrapolation capabilities to higher temperatures, bond breaking, and bond torsions of the acetylacetone molecule was proposed in [3]. In Table 3, we show that MACE achieves state-of-the-art results on this dataset as well. For details, see Appendix 5.3.3.

Table 3: **Root-mean-square errors on the acetylacetone dataset.** Energy (E, meV) and force (F, meV/Å) errors of models trained on configurations of the acetylacetone molecule sampled at 300 K and tested on configurations sampled at 300 K and 600 K. Standard deviations are computed over three runs.

		BOTNet	NequIP	MACE
300 K	E	0.89 (0.0)	**0.81 (0.04)**	0.9 (0.03)
	F	6.3 (0.0)	5.90 (0.38)	**5.1** (0.10)
600 K	E	6.2 (1.1)	6.04 (1.26)	**4.6** (0.3)
	F	29.8 (1.0)	27.8 (3.29)	**22.4** (0.9)
N° Parameters		2,756,416	3,190,488	2,803,984

6 Discussions

With MACE, we extend traditional (equivariant) MPNNs from 2-body to many-body message passing in a computationally efficient manner. Our experiments show that the approach reduces the required

number of message passing, leading to efficient and parallelizable models. Furthermore, we have demonstrated the high accuracy and good extrapolation capabilities of MACE, reaching state-of-the-art accuracy on the rMD17, 3BPA, and AcAc benchmarks. Future development should concentrate on testing MACE on larger systems, including condensed phases and solids.

7 Reproducibility Statements

We have included error bars via different seeds and various ablation studies wherever necessary and appropriate. We have stated all hyper-parameters and data description in the Appendix A.5. Source code is available at `https://github.com/ACEsuit/mace`.

8 Ethical Statements

The societal impact of MACE is challenging to predict. However, better force fields have a positive impact on society by speeding up drug discovery and through helping to understand, control, and design new materials. However, machine learning force fields rely on generating *ab initio* training data leading to heavy computation and large energy consumption. Machine learned force fields do alleviate the costs of doing molecular modelling significantly when compared with using *solely ab initio* methods.

Acknowledgments and Disclosure of Funding

9 Acknowledgement

The authors acknowledge useful discussions with Simon Batzner, Albert Musaelian and William Baldwin. This work was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service (www.csd3.cam.ac.uk), provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/T022159/1), and DiRAC funding from the Science and Technology Facilities Council (www.dirac.ac.uk). DPK acknowledges support from AstraZeneca and the Engineering and Physical Sciences Research Council. CO is supported by Leverhulme Research Project Grant RPG-2017-191 and by the Natural Sciences and Engineering Research Council of Canada (NSERC) [funding reference number IDGR019381].

References

[1] Brandon Anderson, Truong Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/03573b32b2746e6e8ca98b9123f2249b-Paper.pdf`.

[2] Albert P Bartók, Mike C Payne, Risi Kondor, and Gábor Csányi. Gaussian approximation potentials: the accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.*, 104(13): 136403, April 2010.

[3] Ilyes Batatia, Simon Batzner, Dávid Péter Kovács, Albert Musaelian, Gregor N. C. Simm, Ralf Drautz, Christoph Ortner, Boris Kozinsky, and Gábor Csányi. The design space of E(3)-equivariant atom-centered interatomic potentials, 2022. URL `https://arxiv.org/abs/2205.06643`.

[4] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, H. Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew M. Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018. URL `http://arxiv.org/abs/1806.01261`.

[5] 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.

[6] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.*, 98:146401, Apr 2007. doi: 10.1103/PhysRevLett.98.146401. URL https://link.aps.org/doi/10.1103/PhysRevLett.98.146401.

[7] Anton Bochkarev, Yury Lysogorskiy, Christoph Ortner, Gábor Csányi, and Ralf Drautz. Multilayer atomic cluster expansion for semi-local interactions, 2022. URL https://arxiv.org/abs/2205.08177.

[8] Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J Bekkers, and Max Welling. Geometric and physical quantities improve e(3) equivariant message passing, 2021. URL https://arxiv.org/abs/2110.02905.

[9] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021. URL https://arxiv.org/abs/2104.13478.

[10] 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.

[11] Anders S. Christensen and O. Anatole von Lilienfeld. On the role of gradients for machine learning of molecular energies and forces. *Machine Learning: Science and Technology*, 1, 2020. ISSN 23318422. doi: 10.1088/2632-2153/abba6f.

[12] Taco S. Cohen and Max Welling. Group equivariant convolutional networks, 2016. URL https://arxiv.org/abs/1602.07576.

[13] Volker L Deringer, Chris J Pickard, and Gábor Csányi. Data-driven learning of total and local energies in elemental boron. *Physical review letters*, 120(15):156001, 2018.

[14] Volker L Deringer, Miguel A Caro, and Gábor Csányi. A general-purpose machine-learning force field for bulk and nanostructured phosphorus. *Nature communications*, 11(1):1–11, 2020.

[15] Volker L Deringer, Albert P Bartók, Noam Bernstein, David M Wilkins, Michele Ceriotti, and Gábor Csányi. Gaussian process regression for materials and molecules. *Chemical Reviews*, 121(16):10073–10141, 2021.

[16] Ralf Drautz. Atomic cluster expansion for accurate and transferable interatomic potentials. *Phys. Rev. B Condens. Matter*, 99(1):014104, January 2019.

[17] Genevieve Dusson, Markus Bachmayr, Gabor Csanyi, Ralf Drautz, Simon Etter, Cas van der Oord, and Christoph Ortner. Atomic cluster expansion: Completeness, efficiency and stability. *Journal of Computational Physics*, page 110946, 2022.

[18] Felix A. Faber, Anders S. Christensen, Bing Huang, and O. Anatole von Lilienfeld. Alchemical and structural distribution based representation for universal quantum machine learning. *The Journal of Chemical Physics*, 148(24):241717, 2018. doi: 10.1063/1.5020710. URL https://doi.org/10.1063/1.5020710.

[19] Xiang Gao, Farhad Ramezanghorbani, Olexandr Isayev, Justin S. Smith, and Adrian E. Roitberg. Torchani: A free and open source pytorch-based deep learning implementation of the ani neural network potentials. *Journal of Chemical Information and Modeling*, 60(7):3408–3415, 2020. doi: 10.1021/acs.jcim.0c00451. URL https://doi.org/10.1021/acs.jcim.0c00451. PMID: 32568524.

[20] Johannes Gasteiger, Shankari Giri, Johannes T. Margraf, and Stephan Günnemann. Fast and uncertainty-aware directional message passing for non-equilibrium molecules, 2020. URL https://arxiv.org/abs/2011.14115.

[21] Mario Geiger and Tess Smidt. e3nn: Euclidean neural networks, 2022. URL https://arxiv.org/abs/2207.09453.

[22] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry, 2017. URL https://arxiv.org/abs/1704.01212.

[23] Mojtaba Haghighatlari, Jie Li, Xingyi Guan, Oufan Zhang, Akshaya Das, Christopher J. Stein, Farnaz Heidar-Zadeh, Meili Liu, Martin Head-Gordon, Luke Bertels, Hongxia Hao, Itai Leven, and Teresa Head-Gordon. Newtonnet: A newtonian message passing network for deep learning of interatomic potentials and forces, 2021. URL https://arxiv.org/abs/2108.02913.

[24] Mojtaba Haghighatlari, Jie Li, Xingyi Guan, Oufan Zhang, Akshaya Das, Christopher J. Stein, Farnaz Heidar-Zadeh, Meili Liu, Martin Head-Gordon, Luke Bertels, Hongxia Hao, Itai Leven, and Teresa Head-Gordon. Newtonnet: A newtonian message passing network for deep learning of interatomic potentials and forces, 2021.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[26] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically, 2017. URL https://arxiv.org/abs/1712.00409.

[27] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2016. URL https://arxiv.org/abs/1609.02907.

[28] Johannes Klicpera, Shankari Giri, Johannes T. Margraf, and Stephan Günnemann. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. *CoRR*, abs/2011.14115, 2020. URL https://arxiv.org/abs/2011.14115.

[29] Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs, 2020.

[30] Johannes Klicpera, Florian Becker, and Stephan Günnemann. Gemnet: Universal directional graph neural networks for molecules. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=HS_sOaxS9K-.

[31] Risi Kondor. N-body networks: a covariant hierarchical neural network architecture for learning atomic potentials. *arXiv preprint arXiv:1803.01588*, 2018.

[32] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups, 2018. URL https://arxiv.org/abs/1802.03690.

[33] Dávid Péter Kovács, Cas van der Oord, Jiri Kucera, Alice E. A. Allen, Daniel J. Cole, Christoph Ortner, and Gábor Csányi. Linear atomic cluster expansion force fields for organic molecules: Beyond rmse. *Journal of Chemical Theory and Computation*, 17(12):7696–7711, 2021. doi: 10.1021/acs.jctc.1c00647. URL https://doi.org/10.1021/acs.jctc.1c00647. PMID: 34735161.

[34] Yi Liu, Limei Wang, Meng Liu, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d graph networks, 2021. URL https://arxiv.org/abs/2102.05013.

[35] Nicholas Lubbers, Justin S Smith, and Kipton Barros. Hierarchical modeling of molecular energies using a deep neural network. *The Journal of chemical physics*, 148(24):241715, 2018.

[36] 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, 2022. URL https://arxiv.org/abs/2204.05249.

[37] Jigyasa Nigam, Sergey Pozdnyakov, Guillaume Fraux, and Michele Ceriotti. Unified theory of atom-centered representations and message-passing machine-learning schemes. *The Journal of Chemical Physics*, 0(ja):null, 0. doi: 10.1063/5.0087042. URL https://doi.org/10.1063/5.0087042.

[38] Sergey N Pozdnyakov and Michele Ceriotti. Incompleteness of graph convolutional neural networks for points clouds in three dimensions. *arXiv preprint arXiv:2201.07136*, 2022.

[39] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks, 2021. URL https://arxiv.org/abs/2102.09844.

[40] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605.

[41] 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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/303ed4c69846ab36c2904d3ba8573050-Paper.pdf.

[42] Kristof T Schütt, Farhad Arbabzadah, Stefan Chmiela, Klaus R Müller, and Alexandre Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nature communications*, 8(1):1–8, 2017.

[43] Kristof T. Schütt, Oliver T. Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. *CoRR*, abs/2102.03150, 2021. URL https://arxiv.org/abs/2102.03150.

[44] Alexander V. Shapeev. Moment tensor potentials: A class of systematically improvable interatomic potentials. *Multiscale Modeling Simulation*, 14(3):1153–1173, 2016. doi: 10.1137/15M1054183. URL https://doi.org/10.1137/15M1054183.

[45] Philipp Thölke and Gianni De Fabritiis. Equivariant transformers for neural network based molecular potentials. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=zNHzqZ9wrRB.

[46] Nathaniel Thomas, Tess Smidt, Steven M. Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds. *CoRR*, abs/1802.08219, 2018. URL http://arxiv.org/abs/1802.08219.

[47] A.P. Thompson, L.P. Swiler, C.R. Trott, S.M. Foiles, and G.J. Tucker. Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials. *Journal of Computational Physics*, 285:316–330, 2015. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2014.12.018. URL https://www.sciencedirect.com/science/article/pii/S0021999114008353.

[48] Oliver T Unke and Markus Meuwly. Physnet: A neural network for predicting energies, forces, dipole moments, and partial charges. *Journal of chemical theory and computation*, 15(6): 3678–3693, 2019.

[49] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. In *NeurIPS*, pages 10402–10413, 2018. URL http://papers.nips.cc/paper/8239-3d-steerable-cnns-learning-rotationally-equivariant-features-in-volumetric-data.

[50] Michael J Willatt, Félix Musil, and Michele Ceriotti. Feature optimization for atomistic machine learning yields a data-driven construction of the periodic table of the elements. *Physical Chemistry Chemical Physics*, 20(47):29661–29668, 2018.

[51] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications, 2018. URL https://arxiv.org/abs/1812.08434.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyper-parameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

A.1 Acetylacetone Dataset: Additional Experiments

We ran additional experiments with the acetylacetone dataset introduced in [3] to further investigate the generalization capabilities of MACE [3]. Figure 4 shows the energy predictions of BOTNet [3], NequIP [5], MACE, and (linear) ACE [33] for two trajectories on the acetylacetone's potential energy surface (PES). The left panel shows the energy profile for a rotation around an O-C-C-C dihedral angle. Since the training set only contains dihedral angles below 30° (see lower panel), accurate predictions for angles up to 180° require significant extrapolation capabilities. Also the energy barrier of the rotation is with 1 eV well outside the energy range of the training set which is sampled at 300 K. It can be seen that all models solve this task surprisingly well.

In the right panel of Figure 4, we show energy predictions along a minimum energy path of an intramolecular hydrogen transfer reaction. This task probes a model's ability to describe a bond breaking reaction, something it has not seen in the training data. It should be noted that this reaction occurs in a region of the PES that is not too far from the training data as can be seen from the histogram below. All models accurately reproduce the barrier's shape with the MPNN models closely matching the barrier height as well.

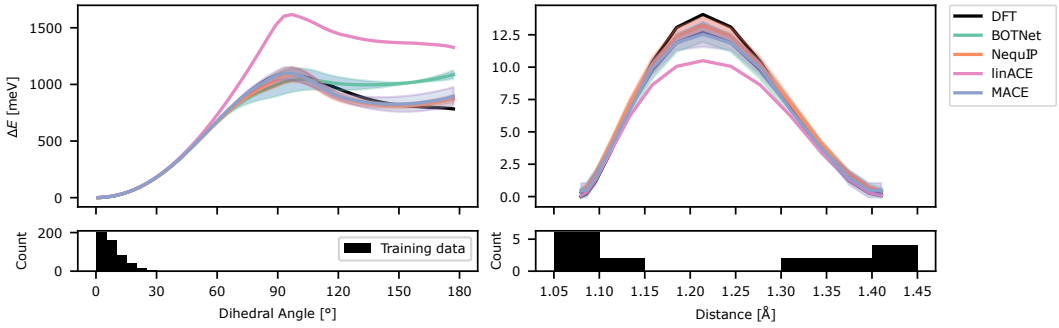

Figure 4: *Left:* energy predictions for a dihedral slice of the DFT potential energy surface of acetylacetone. *Right:* energy predictions for the proton transfer in acetylacetone. Error bars indicate one standard deviation computed over three runs. The histograms show the distribution of the training data along the relevant coordinate.

A.2 Description of the Datasets

A.2.1 rMD17 Dataset

The revised MD17 (rMD17) dataset contains five train test splits of 10 different small organic molecules [11]. Each of the splits contains 1000 configurations for each molecule sampled randomly from a long *Ab initio* molecular dynamics simulation carried out at 500 K computed with DFT. We note that the older version of the dataset, called MD17, has been shown to contain noisy labels [11].

A.2.2 3BPA Dataset

The 3BPA dataset contains DFT train test splits of a flexible drug-like organic molecule sampled from different temperature molecular dynamics trajectories [33]. The models is trained on 500 snapshots sampled at 300K and tested on three independent test sets for each temperature (300K, 600K, 1200K). The models can also be tested on the challenging task of computing the energy along dihedral rotations of the molecule. This test directly probes the smoothness and accuracy of the part of PES that determines which conformers are present in a simulation, and hence has a direct influence on properties of interest such as binding free energies to protein targets.

A.2.3 Acetylacetone Dataset

The acetylacetone dataset contains trajectories of a small reactive molecule sampled at different temperature. The task is to train on snapshot sampled at 300K and test on independent test sets sampled at 300K and 600K. Moreover the extrapolation is measured both in temperature and along two internal coordinates of the molecule, the hydrogen transfer path and a partially conjugated double bond rotation, which has a very high barrier for rotation.

A.3 Implementation Details

A.3.1 Symmetrised One-particle Basis

For the implementation of the one-particle basis, we use `e3nn` [21] for the spherical harmonics and for symmetrising the tensor product of (8). Consequently, we also use their internal normalization.

A.3.2 Generalized Clebsch-Gordan Coefficients

The generalised Clebsch-Gordan coefficients are defined as product of Clebsch-Gordan coefficients:

$$\mathcal{C}^{LM}_{l_1 m_1,..,l_n m_n} = C^{L_2 M_2}_{l_1 m_1, l_2 m_2} C^{L_3 M_3}_{L_2 M_2, l_3 m_3} ... C^{L_N M_N}_{L_{N-1} M_{N-1}, l_N m_N}, \tag{14}$$

where $L \equiv (L_2, .., L_N)$, $|l_1 - l_2| \leq L_2 \leq l_1 + l_2 \; \forall \; i \geq 3 | L_{i-1} - l_i| \leq L_i \leq L_{i-1} + l_i$, and $M_i \in \{m_i | - l_i \leq m_i \leq l_i\}$.

A.3.3 Higher Order Features Via Loop Tensor Contractions

We implement the construction of the higher order features of Equation (10) and the message of Equation (11) in a single efficient loop tensor contraction algorithm. Below, we drop the t superscript for clarity. The input variables are defined in Section 4. We give here a brief reminder, along with shape of the tensors to contract.

Algorithm 1 Efficient implementation of (10) and (11) through tensor contractions of A-features $A_{i,klm}$ of size $[N_{\text{atoms}}, N_{\text{channels}}, (l_{\max}+1)^2]$, generalized Clebsch-Gordan coefficients $\mathcal{C}^{LM}_{\eta,l_1 m_1,...,l_{\tilde{\nu}} m_{\tilde{\nu}}}$ of size $[N_{\text{coupling},\tilde{\nu}}] \times [(l_{\max}+1)^2]^{\tilde{\nu}}$, and weights $W_{z_i kL,\eta_{\tilde{\nu}}}$ of size $[N_{\text{elements}}, N_{\text{channels}}, N_{\text{coupling},\tilde{\nu}}]$ for a given correlation order $\tilde{\nu}$. $l_{\max}$ is the highest symmetry order of the spherical expansion of the A-features and L is the targeted symmetry order. N_{coupling} is the number of couplings of $l_1 \dots l_{\tilde{\nu}}$ that yield L. In practice, we vectorize over $M \in \{-L, -L+1, ..., L-1, L\}$.

function LOOPEDTENSORCONTRACTION($A_{i,klm}$, $\{W_{z_i kL,\eta_{\tilde{\nu}}}\}_{\tilde{\nu} \leq \nu}$, $\{\mathcal{C}^{LM}_{\eta,l_1 m_1,...,l_{\tilde{\nu}} m_{\tilde{\nu}}}\}_{\tilde{\nu} \leq \nu}$, ν)

$\tilde{c}^{z_i kLM}_{l_1 m_1,...,l_\nu m_\nu} \leftarrow \sum_\eta \mathcal{C}^{LM}_{\eta,l_1 m_1,...,l_\nu m_\nu} W_{z_i kL,\eta_\nu}$ ▷ Contract coupling coefficients and weights

$a^{z_i kLM}_{i,l_1 m_1,...,l_{\nu-1} m_{\nu-1}} \leftarrow \sum_{l_\nu m_\nu} \tilde{c}^{z_i kLM}_{l_1 m_1,...,l_\nu m_\nu} A_{i,kl_\nu m_\nu}$

for $\tilde{\nu} \leftarrow \nu - 1$ to 1 **do** ▷ Iterate over correlation orders

$\tilde{c}^{z_i kLM}_{l_1 m_1,...,l_{\tilde{\nu}} m_{\tilde{\nu}}} \leftarrow \sum_\eta \mathcal{C}^{LM}_{\eta,l_1 m_1,...,l_{\tilde{\nu}} m_{\tilde{\nu}}} W_{z_i kL,\eta_{\tilde{\nu}}}$

$\tilde{a}^{z_i kLM}_{i,l_1 m_1,...,l_{\tilde{\nu}} m_{\tilde{\nu}}} \leftarrow a^{z_i kLM}_{i,l_1 m_1,...,l_{\tilde{\nu}} m_{\tilde{\nu}}} + \tilde{c}^{z_i kLM}_{l_1 m_1,...,l_{\tilde{\nu}} m_{\tilde{\nu}}}$

$a^{z_i kLM}_{i,l_1 m_1,...,l_{\tilde{\nu}-1} m_{\tilde{\nu}-1}} \leftarrow \sum_{l_{\tilde{\nu}} m_{\tilde{\nu}}} \tilde{a}^{z_i kLM}_{i,l_1 m_1,...,l_{\tilde{\nu}} m_{\tilde{\nu}}} A_{i,kl_{\tilde{\nu}} m_{\tilde{\nu}}}$

end for

return $a^{z_i kLM}_i$ ▷ Return message $m_{i,kLM}$

end function

The algorithm starts at correlation ν. The first step of the algorithm is to contract the generalized Clebsch-Gordan coefficients with the weights of the product basis. This contractions trades the computational cost of several products for that of a sum which is computationally very advantageous. Then, the last dimension of $\tilde{c}_\nu$ is contracted with the A_i-features' last dimension resulting in the a-tensor with correlation order $\nu - 1$. The algorithm then loops over the correlation order $\tilde{\nu}$ in descending order until $\tilde{\nu} = 1$. In each step, we first create the tensor $\tilde{c}$ by contracting the Clebsch-Gordan coefficients of correlation order $\tilde{\nu}$ with the weights. Then, the previous a-tensor is added to

the $\tilde{c}$-tensor. This operation ensures that at the end of the loop, the product basis of every correlation order are created. In fact, the a contains the products for ν to $\tilde{\nu}$. The updated tensor is then contracted again with the atomic basis increasing the correlation order by 1 for all the products presented in a. The last a tensor is exactly the message of (11).

A.4 Tensor shapes

A glossary of the shapes of the various tensors in the MACE architecture.

Tensor	Shapes	Equation
$h^{(t)}_{i,kl_2m_2}$	[N_atoms, N_channels, $(l_2^{\max}+1)^2$]	(8)
$R^{(t)}_{kl_1l_2l_3}(r_{ji})$	[N_edges, N_channels, N_basis]	(8)
$C^{l_3m_3}_{l_1m_1,l_2m_2}$	$[2\times l_3+1, 2\times l_1+1, 2\times l_2+1]$	(8)
$A^{(t)}_{i,kl_3m_3}$	[N_atoms, N_channels, $(l_3^{\max}+1)^2$]	(8)
$\mathcal{C}^{LM}_{\eta_\nu,\mathbf{lm}}$	$[(2\times L+1), [(l^{\max}+1)^2]^\nu$, N_path]	(10)
$\boldsymbol{B}^{(t)}_{i,\eta_\nu kLM}$	[N_atoms, N_channels, N_path, $(2\times L+1)$]	(10)
$W_{z_i kL,\eta_\nu}$	[N_channels, N_elements, N_path]	(10)
$m^{(t)}_{i,kLM}$	[N_atoms, N_channels, $(L+1)^2$]	(11)

A.5 Training Details

We used three codes for the paper. All MACE experiments were run with the `mace` code. All BOTNet experiments were run within the `mace` code. For NequIP experiments, we detail hereafter what code was used for what experiment. We train with `float64` precision for 3BPA and AcAc and `float32` precision for rMD17.

A.5.1 MACE

Models were trained on an NVIDIA A100 GPU in single GPU training. Typical training time for MACE models is between 2 to 6 hours depending on the dataset. The revised MD17 models were trained with a total budget of 1,000 configurations, split into 950 for training and 50 for validation. The 3BPA models were trained on 500 structures, split into 450 for training and 50 for validation. The AcAc models were trained on 500 structures, split into 450 for training and 50 for validation. The data set was reshuffled after each epoch. We use two layers and 256 uncoupled feature channels and $l_{\max}=3$. For all models, radial features are generated using 8 Bessel basis functions and a polynomial envelope for the cutoff with $p=5$ [29]. The radial features are fed to an MLP of size [64, 64, 64, 1024], using SiLU nonlinearities on the outputs of the hidden layers. The readout function of the first layer is implemented as a simple linear transformation. The readout function of the second layer is a single-layer MLP with 16 hidden dimensions. We used a 5 Å cutoff for all molecules. We use the following loss function:

$$\mathcal{L}=\frac{\lambda_E}{B}\sum_{b}^{B}\left(\hat{E}_b-E_b\right)^2+\frac{\lambda_F}{3BN}\sum_{i=1}^{B\cdot N}\sum_{\alpha=1}^{3}\left(-\frac{\partial\hat{E}}{\partial r_{i,\alpha}}-F_{i,\alpha}\right)^2, \tag{15}$$

where B denotes the number of batches, N the number of atoms in the batch, E_b the ground-truth energy, $\hat{E}_b$ the predicted energy, $F_{i,\alpha}$ the force component of atom i in the direction $\alpha\in\{\hat{x},\hat{y},\hat{z}\}$. λ_E and λ_F are weights set to 1 and $1,000$, respectively.

Models were trained with AMSGrad variant of Adam, with default parameters of β_1 = 0.9, β_2 = 0.999, and $\epsilon=10^{-8}$. We used a learning rate of 0.01 and a batch size of 5. The learning rate was reduced using an on-plateau scheduler based on the validation loss with a patience of 50 and a decay factor of 0.8. We use an exponential moving average with weight 0.99 to evaluate on the validation set as well as for the final model, an exponential weight decay of 5e^{-7} on the weights of equation 10 and 11, and a per-atom shift via the average per-atom energy over all the training set and a per-atom scale as the root mean-square of the components of the forces over the training set.

A.5.2 NequIP

We use two implementations of NequIP for the results in the paper. We trained models on NVIDIA A100 GPU in single GPU training. Typical training time for NequIP models is between 6 hours to 2 days depending on the dataset. The results for 50 configurations rMD17 molecule were done using nequip code. The models with increasing number of layers was trained in the mace code. The timings for NequIP were also done in the mace code.

Original `nequip` code base [5] The NequIP model was trained on 50 configurations of rMD17 used the same model specifications as for rMD17 in [5] with the same training procedure. λ_E and λ_F were set to 1 and $1,000$, respectively.

Reimplementation of NequIP in the `mace` code base For the increasing layer experiment, the NequIP model was trained on 450 configurations and 50 configs were used for validation. We use 5 layers with 64 channels for even and odd parity, and $L = 3$ messages. We use a cutoff radius of 4Å. Radial features are generated using 8 Bessel basis functions and a polynomial envelope for the cutoff with p = 6. We use λ_E and λ_F weights set to 1 and $1,000$, respectively. Models were trained with AMSGrad variant of Adam, with default parameters of β_1 = 0.9, β_2 = 0.999, and $\epsilon = 10^{-8}$. We used a learning rate of 0.01 and a batch size of 5. The learning rate was reduced using an on-plateau scheduler based on the validation loss with a patience of 50 and a decay factor of 0.8. We use an exponential moving average with weight 0.99 to evaluate on the validation set as well as for the final model.

For the 3BPA timings model, we use the same model specification as in [36] with the important difference of using a polynomial envelope for the cutoff with $p = 6$ instead of $p = 2$. .