Higher-Rank Irreducible Cartesian Tensors for Equivariant Message Passing

Viktor Zaverkin[1,*] **Francesco Alesiani**[1,†] **Takashi Maruyama**[1,†] **Federico Errica**[2]
Henrik Christiansen[1] **Makoto Takamoto**[1] **Nicolas Weber**[1] **Mathias Niepert**[1,3]
[1]NEC Laboratories Europe [2]NEC Italy [3]University of Stuttgart

Abstract

The ability to perform fast and accurate atomistic simulations is crucial for advancing the chemical sciences. By learning from high-quality data, machine-learned interatomic potentials achieve accuracy on par with ab initio and first-principles methods at a fraction of their computational cost. The success of machine-learned interatomic potentials arises from integrating inductive biases such as equivariance to group actions on an atomic system, e.g., equivariance to rotations and reflections. In particular, the field has notably advanced with the emergence of equivariant message passing. Most of these models represent an atomic system using spherical tensors, tensor products of which require complicated numerical coefficients and can be computationally demanding. Cartesian tensors offer a promising alternative, though state-of-the-art methods lack flexibility in message-passing mechanisms, restricting their architectures and expressive power. This work explores higher-rank irreducible Cartesian tensors to address these limitations. We integrate irreducible Cartesian tensor products into message-passing neural networks and prove the equivariance and traceless property of the resulting layers. Through empirical evaluations on various benchmark data sets, we consistently observe on-par or better performance than that of state-of-the-art spherical and Cartesian models.

1 Introduction

The ability to perform sufficiently fast and accurate atomistic simulations for large molecular and material systems holds the potential to revolutionize molecular and materials science [1–8]. Conventionally, computational chemistry and materials science rely on ab initio or first-principles approaches—e.g., coupled cluster [9–11] or density functional theory (DFT) [12, 13], respectively—that are accurate but computationally demanding, thus limiting the accessible simulation time and system sizes. However, the ability to generate high-quality, first-principles-based data sets has prompted the development of machine-learned interatomic potentials (MLIPs). These potentials enable atomistic simulations with accuracy that is on par with the reference first-principles method but at a fraction of the computational cost. Message-passing neural networks (MPNNs) [14–21] have been employed in chemical and materials sciences, including the development of MLIPs, due to their efficient processing of the graph representation of the atomic system [22–26]. Achieving the desired MLIPs' performance, however, requires the inclusion of inductive biases, like the invariance of total energy to translations, reflections, and rotations in the three-dimensional space, and an effective encoding of the atomic system into a learnable representation [27]. Designing equivariant MPNNs [25, 26, 28–36], which preserve the directional information of the local atomic environment, has been one of the most active research directions of the last years. Previous work often achieves equivariance by representing an atomic

[*]Corresponding author: viktor.zaverkin@neclab.eu
[†]These authors contributed equally.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-3940

system as a graph, with node features expressed in spherical harmonics basis or using lower-rank Cartesian tensors and designing ad-hoc message-passing layers [25, 28–37].

MLIPs based on spherical harmonics [25, 33, 34], which are the basis for irreducible representations of the three-dimensional rotation group, often demonstrate a better performance compared to those that use lower-rank Cartesian representations (scalars and vectors) [28, 37]. Spherical tensors, however, require the definition of a particular rotational axis, often chosen as the z-axis, resulting in inherent bias [38]. Furthermore, coupling spherical tensors via tensor products, involved in designing equivariant convolutions and constructing many-body features [33, 39, 40], requires the definition of complicated numerical coefficients, e.g., Wigner 3-j symbols defined in terms of the Clebsch–Gordan coefficients [41], and can be computationally demanding. In contrast, irreducible Cartesian tensors have no preferential directions; their tensor products are simpler and have a better computational complexity—up to a certain tensor rank—than the tensor products of spherical tensors [38, 42–45]. Recent work improved results of Cartesian MPNNs by using rank-two tensors and decomposing them into irreducible representations of the three-dimensional rotation group [30], i.e., into representations of dimension 1 (trace), 3 (anti-symmetric part), and 5 (traceless symmetric part) [46]. Higher-rank reducible Cartesian tensors have also been integrated into many-body MPNNs [31]. These state-of-the-art Cartesian approaches, however, lack the flexibility of their message-passing mechanisms. They rely exclusively on convolutions with invariant filters and restrict the construction of many-body features, limiting the range of possible architectures and their expressive power.

Contributions. In this work, we address the limitations of state-of-the-art Cartesian models and demonstrate that operating with irreducible Cartesian tensors leads to on-par or, sometimes, even better performance than that of spherical counterparts. Particularly, this work goes beyond scalars, vectors, and rank-two tensors and explores the integration of higher-rank irreducible Cartesian tensors and their products into equivariant MPNNs: i) We demonstrate how irreducible Cartesian tensors that are symmetric and traceless can be constructed from a unit vector and a product of two irreducible Cartesian tensors, essential for equivariant convolutions and constructing many-body features; ii) We prove that the resulting tensors are traceless and equivariant under the action of the three-dimensional rotation and reflection group; iii) We demonstrate that higher-rank irreducible Cartesian tensors can be used to design cost-efficient—up to a certain tensor rank—and accurate equivariant models of many-body interactions; iv) We conduct different experiments to assess the effectiveness of equivariant message passing based on irreducible Cartesian tensors, and achieve state-of-the-art performance on benchmark data sets such as rMD17 [47], MD22 [48], 3BPA [49], acetylacetone [32], and Ta–V–Cr–W [50]. We hope our contributions will offer new insights into the use of Cartesian tensors for constructing accurate and cost-efficient MLIPs and beyond.

2 Background

Group representations and equivariance. A group $(G, \cdot)$ is defined by a set of elements G and a group product $\cdot$. A representation D of a group is a function from G to square matrices such that $D[g]D[g'] = D[g \cdot g']$, $\forall\, g, g' \in G$. This representation defines an action of G to any vector space $\mathcal{X}$ (of the same dimension as the dimension of square matrices) through the matrix-vector multiplication, i.e., $(g, \mathbf{x}) \mapsto D_{\mathcal{X}}[g]\mathbf{x}$ for $\forall\, g \in G$ and $\forall\, \mathbf{x} \in \mathcal{X}$. For vector spaces $\mathcal{X}$ and $\mathcal{Y}$, a function $f : \mathcal{X} \to \mathcal{Y}$ is called equivariant to the action of a group G to $\mathcal{X}$ and $\mathcal{Y}$ iff

$$f(D_{\mathcal{X}}[g]\mathbf{x}) = D_{\mathcal{Y}}[g] f(\mathbf{x}) \,, \forall\, g \in G.$$

Invariance can be seen as a special type of equivariance with $D_{\mathcal{Y}}[g]$ being the identity for all g. An important class of equivariant models focuses on equivariance to the action of the Euclidean group $\mathrm{E}(3)$, comprising translations and the orthogonal group $\mathrm{O}(3)$, i.e., rotation group $\mathrm{SO}(3)$ and reflections, in $\mathbb{R}^3$. MLIPs based on equivariant models usually focus on equivariance to the action of the orthogonal group $\mathrm{O}(3)$ and are invariant to translations.

Cartesian tensors. A vector $\mathbf{x} \in \mathbb{R}^3$ transforms under the action of the rotation group $\mathrm{SO}(3)$ as $\mathbf{x}' = R\mathbf{x}$, i.e., each component of it transforms as $(\mathbf{x})'_i = \sum_j R_{ij} (\mathbf{x})_j$. Here, $R = D_{\mathcal{X}}[g] \in \mathbb{R}^{3\times3}$ denotes the rotation matrix representation of $g \in \mathrm{SO}(3)$. Cartesian tensors of rank n are described by 3^n numbers and generalize the concept of vectors. A rank-n Cartesian tensor $\mathbf{T}$ can be viewed as a multidimensional array with n indices, i.e., $(\mathbf{T})_{i_1 i_2 \cdots i_n}$ with $i_k \in \{1, 2, 3\}$ for $\forall\, k \in \{1, \cdots, n\}$. Furthermore, each index of $(\mathbf{T})_{i_1 i_2 \cdots i_n}$ transforms independently as a vector under rotation. For example, a rank-two tensor transforms under rotation as $\mathbf{T}' = R\mathbf{T}R^\top$, i.e., each component of

it transforms as $(\mathbf{T})'_{i_1 i_2} = \sum_{j_1 j_2} R_{i_1 j_1} R_{i_2 j_2} (\mathbf{T})_{j_1 j_2}$. For Cartesian tensors, one defines an r-fold tensor contraction and an outer product, i.e., $(\mathbf{T})_{j_1 \cdots j_s} = \sum_{i_1, \cdots, i_r} (\mathbf{U})_{i_1 \cdots i_r j_1 \cdots j_s} (\mathbf{S})_{i_1 \cdots i_r}$ and $(\mathbf{T})_{i_1 \cdots i_r j_1 \cdots j_s} = (\mathbf{U})_{i_1 \cdots i_r} (\mathbf{S})_{j_1 \cdots j_s}$, respectively. Cartesian tensors are generally reducible and can, thus, be decomposed into smaller representations that transform independently within their linear subspaces under rotation. For example, a rank-two reducible Cartesian tensor contains representations of dimension 1 (trace), 3 (anti-symmetric part), and 5 (traceless symmetric part): $\sum_{i_1} (\mathbf{T})_{i_1 i_1}$, $(\mathbf{T})_{i_1 i_2} - (\mathbf{T})_{i_2 i_1}$, and $(\mathbf{T})_{i_1 i_2} + (\mathbf{T})_{i_2 i_1} - 2/3 \sum_{i_1} (\mathbf{T})_{i_1 i_1}$, respectively. Under rotation, the traceless symmetric part remains within its irreducible subspace, with a similar behavior for other irreducible components of $(\mathbf{T})_{i_1 i_2}$. Furthermore, for a reducible Cartesian tensor of rank n, an action of the rotation group $\mathrm{SO}(3)$ can be represented with a $3^n \times 3^n$-dimensional rotation matrix. This rotation matrix is also reducible, meaning that an appropriate change of basis can block diagonalize it into smaller subsets, the irreducible representations of the rotation group $\mathrm{SO}(3)$.

Spherical tensors. Spherical harmonics (or spherical tensors) Y_m^l are functions from the points on a sphere to complex or real numbers, with degree $l \geq 0$ and components $-l \leq m \leq l$. Collecting all components for a given l we obtain a $(2l+1)$-dimensional object $\mathbf{Y}^l = (Y_{-l}^l, \ldots, Y_{l-1}^l, Y_l^l)$. Spherical tensors transform under rotation as $Y_m^l(R\hat{\mathbf{x}}) = \sum_{m'} (\mathbf{D})_{mm'}^l Y_{m'}^l(\hat{\mathbf{x}})$, where $(\mathbf{D})_{mm'}^l$ is the irreducible representation of $\mathrm{SO}(3)$—the Wigner D-matrix—and R denotes the rotation matrix. Spherical tensors are irreducible and can be combined using the Clebsch–Gordan tensor product

$$\left(Y_{m_1}^{l_1} \otimes_{\mathrm{CG}} Y_{m_2}^{l_2}\right)_{m_3}^{l_3} = \sum\nolimits_{m_1=-l_1}^{l_1} \sum\nolimits_{m_2=-l_2}^{l_2} C_{l_1 m_1, l_2 m_2}^{l_3 m_3} Y_{m_1}^{l_1} Y_{m_2}^{l_2},$$

with Clebsch–Gordan coefficients $C_{l_1 m_1, l_2 m_2}^{l_3 m_3}$ and $l_3 \in \{|l_1 - l_2|, \cdots, l_1 + l_2\}$. Furthermore, any reducible Cartesian tensor of rank n can be written in spherical harmonics as $(\hat{\mathbf{x}})_{i_1} (\hat{\mathbf{x}})_{i_2} \cdots (\hat{\mathbf{x}})_{i_n} = \sum_{l=0}^{n} \sum_{m=-l}^{l} X_{i_1 i_2 \cdots i_n}^{lm} Y_m^l$, with coefficients $X_{i_1 i_2 \cdots i_n}^{lm}$ defined elsewhere [51].

3 Related work

Equivariant message-passing potentials. Equivariant MPNNs [25, 28–37, 39, 52–54] often outperform more traditional invariant models [55–60]. While many invariant and equivariant MPNNs rely on two-body features within a single message-passing layer, there is a growing interest in constructing higher-body-order features, such as angles and dihedrals, to model many-body interactions in atomic systems [57, 59, 60]. Note that equivariant models with two-body features in a single message-passing layer build many-body ones through repeated message-passing iterations, increasing the receptive field and, thus, computational cost. Recent work recognized the importance of higher-body-order features but faced challenges due to explicit summation over triplets or quadruplets [57, 59]. In contrast, MACE advances the current state-of-the-art by combining ACE and equivariant message passing, introducing cost-efficient many-body message-passing potentials [33]. With just two message-passing layers, MACE yields accurate potentials for interacting many-body systems [61].

Beyond Clebsch–Gordan tensor product. Despite the success of equivariant MPNNs based on spherical tensors, the high computational cost of the Clebsch–Gordan tensor product limits their computational efficiency [28, 30, 33–35, 53, 54, 62–64]. Thus, current research focuses on alternatives to the Clebsch–Gordan tensor product, which has a $\mathcal{O}(L^5)$ complexity for tensors up to degree L. Recently, the relation of Clebsch–Gordan coefficients to the integral of products of three spherical harmonics, known as the Gaunt coefficients [41], has been exploited to reduce the computational cost of the tensor product of spherical tensors to $\mathcal{O}(L^3)$, compared to $\mathcal{O}(L^6)$ of the full Clebsch–Gordan tensor product including all $(l_1, l_2) \to l_3$ [36]. However, this approach excludes odd tensor products and, thus, restricts the expressive power and the range of possible architectures.

Cartesian tensors and their products offer another promising alternative to the Clebsch–Gordan tensor product, enabling the efficient construction of message-passing layers with two- and many-body features. Recent work has explored decomposing rank-two tensors into their irreducible representations and using higher-rank reducible tensors [30, 31]. These approaches, however, are limited to convolutions with invariant filters, restricting the range of possible architectures and, thus, the expressive power of resulting Cartesian models. Furthermore, they provide limited mechanisms for constructing higher-body-order features, restricted to three-body features obtained through matrix-matrix multiplication and invariant many-body features obtained through full tensor contractions, similar to moment tensor potentials and Gaussian moments [65–67]. Finally, using reducible Cartesian

 https://doi.org/10.52202/079017-3940

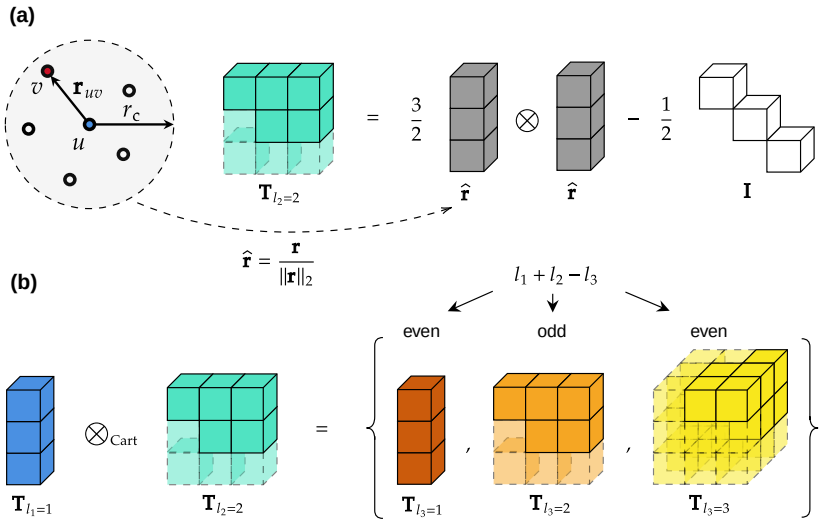

Figure 1: **Schematic illustration of (a) the construction of an irreducible Cartesian tensor for a local atomic environment and (b) the tensor product of two irreducible Cartesian tensors of rank l_1 and l_2.** The construction of an irreducible Cartesian tensor from a unit vector $\hat{\mathbf{r}}$ is defined in Eq. (1). In this work, we use tensors with the same rank n and weight l, i.e., $n = l$, avoiding the need for embedding tensors with $l < n$ in a higher-dimensional tensor space. Therefore, we use l to identify the rank and the weight of an irreducible Cartesian tensor. The tensor product is defined in Eqs. (2) and (3), resulting in a new tensor $\mathbf{T}_{l_3} = (\mathbf{T}_{l_1} \otimes_{\text{Cart}} \mathbf{T}_{l_2})_{l_3}$ of rank $l_3 = \{|l_1 - l_2|, \cdots, l_1 + l_2\}$. Transparent boxes denote the linearly dependent elements of symmetric and traceless tensors. The tensor product can be even or odd, defined by $l_1 + l_2 - l_3$.

tensors during message-passing leads to mixing different irreducible representations, which can hinder the performance of resulting models compared to state-of-the-art spherical models [31].

4 Methods

We define an atomic configuration $\mathcal{S} = \{\mathbf{r}_u, Z_u\}_{u=1}^{N_{\text{at}}}$, where $\mathbf{r}_u \in \mathbb{R}^3$ denotes Cartesian coordinates and $Z_u \in \mathbb{N}$ represents the atomic number of atom u, with a total of N_{at} atoms. Our focus lies on message-passing MLIPs, parameterized by $\boldsymbol{\theta}$, that learn a mapping from a configuration $\mathcal{S}$ to the total energy E, i.e., $f_{\boldsymbol{\theta}} : \mathcal{S} \mapsto E \in \mathbb{R}$. Thus, we represent molecular and material systems as graphs in a three-dimensional Euclidean space. An edge $\{u, v\}$ exists if atoms u and v are within a cutoff distance r_c, i.e., $\|\mathbf{r}_u - \mathbf{r}_v\|_2 \leq r_c$. For more details on MPNNs, see Appendix A. The total energy of an atomic configuration is defined by the sum of individual atomic energy contributions [68], i.e., $E = \sum_{u=1}^{N_{\text{at}}} E_u$. Atomic forces are computed as negative gradients of the total energy with respect to atomic coordinates, i.e., $\mathbf{F}_u = -\nabla_{\mathbf{r}_u} E$.

4.1 Irreducible Cartesian tensor product

In the following, we explore irreducible Cartesian tensors and their products based on the three-dimensional vector space and the three-dimensional orthogonal group $\mathrm{O}(3)$, which comprises rotations and reflections. The respective tensors and tensor products are schematically illustrated in Fig. 1 (a) and (b), respectively, and are further employed in constructing MPNNs for atomic systems equivariant under actions of the orthogonal group. An irreducible Cartesian tensor $\mathbf{T}_n \in (\mathbb{R}^3)^{\otimes n}$ of rank n and weight $l \leq n$ (related to the degree l of spherical tensors) can be represented by a tensor with 3^n components in a three-dimensional vector space. These 3^n components form a basis for a $(2l+1)$-dimensional irreducible representation of the three-dimensional rotation group $\mathrm{SO}(3)$ [69, 70]. Thus, only $2l + 1$ of the 3^n components are independent.

The rotation in the space of all tensors of rank n is induced through the n-fold outer product of a rotation matrix $R \in \mathbb{R}^{3\times3}$, i.e., $R^{\otimes n} = R \otimes \cdots \otimes R$. The obtained representation of the

rotation group $\mathrm{SO}(3)$ is reducible for all n except $n = \{0, 1\}$. Reducing a tensor of rank n yields a unique irreducible tensor with the same weight and rank ($n = l$), which is characterized by being symmetric, i.e., $(\mathbf{T}_n)_{\dots i \dots j \dots} = (\mathbf{T}_n)_{\dots j \dots i \dots}, \forall\, i \neq j \in \{i_1 \cdots, i_n\}$, and traceless, i.e., $\sum_i (\mathbf{T}_n)_{\dots i \dots i \dots} = 0\,, \forall\, i \in \{i_1, \cdots, i_n\}$. An irreducible tensor of rank n and weight l with $l < n$ can be viewed as a l-rank tensor embedded in the n-rank tensor space, e.g., by computing an outer product with the identity matrix. However, the embedding is often not unique. Thus, we construct tensors with the same weight and rank ($n = l$) in the following.

Irreducible Cartesian tensors from unit vectors. Hereafter, we denote the rank and the weight of irreducible Cartesian tensors by l to distinguish them from reducible counterparts. An irreducible Cartesian tensor of an arbitrary rank l can be constructed from a unit vector $\hat{\mathbf{r}}$ in the form [45]

$$\mathbf{T}_l(\hat{\mathbf{r}}) = C \sum\nolimits_{m=0}^{\lfloor l/2 \rfloor} (-1)^m \frac{(2l-2m-1)!!}{(2l-1)!!} \big\{ \hat{\mathbf{r}}^{\otimes(l-2m)} \otimes \mathbf{I}^{\otimes m} \big\}, \tag{1}$$

resulting in a symmetric and traceless tensor of rank l. Here, $\mathbf{I}$ denotes the 3×3 identity matrix, $\hat{\mathbf{r}}^{\otimes(l-2m)} = \hat{\mathbf{r}} \otimes \cdots \otimes \hat{\mathbf{r}}$ and $\mathbf{I}^{\otimes m} = \mathbf{I} \otimes \cdots \otimes \mathbf{I}$ are the corresponding $(l-2m)$- and m-fold outer products. The curly brackets indicate the summation over all permutations of the l unsymmetrized indices [45], i.e., $\{\mathbf{T}_l\}_{i_1 \cdots i_l} = \sum_{\pi \in S_l} (\mathbf{T}_l)_{i_{\pi(1)} \cdots i_{\pi(l)}}$, with S_l being the corresponding set of permutations. The expression in Eq. (1) involves three distinct outer products $(\hat{\mathbf{r}} \otimes \hat{\mathbf{r}})_{i_1 i_2} = \hat{r}_{i_1} \hat{r}_{i_2}$, $(\mathbf{I} \otimes \mathbf{I})_{i_1 i_2 i_3 i_4} = \delta_{i_1 i_2} \delta_{i_3 i_4}$, and $(\hat{\mathbf{r}} \otimes \mathbf{I})_{i_1 i_2 i_3} = \hat{r}_{i_1} \delta_{i_2 i_3}$, where $\delta_{i_2 i_3}$ denotes the Kronecker delta. The normalization constant $C = (2l-1)!!/l!$ is chosen such that an l-fold contraction of $\mathbf{T}_l$ with the unit vector $\hat{\mathbf{r}}$ yields unity. An example of an irreducible Cartesian tensor with rank $l = 3$ is $(\mathbf{T}_{l=3})_{i_1 i_2 i_3} = \frac{5}{2} \big(\hat{r}_{i_1} \hat{r}_{i_2} \hat{r}_{i_3} - \frac{1}{5} (\hat{r}_{i_1} \delta_{i_2 i_3} + \hat{r}_{i_2} \delta_{i_3 i_1} + \hat{r}_{i_3} \delta_{i_1 i_2}) \big)$.

Irreducible Cartesian tensor product. The following defines the product of two irreducible Cartesian tensors $\mathbf{x}_{l_1} \in (\mathbb{R}^3)^{\otimes l_1}$ and $\mathbf{y}_{l_2} \in (\mathbb{R}^3)^{\otimes l_2}$, yielding an irreducible Cartesian tensor of rank l_3, i.e., $\mathbf{z}_{l_3} = (\mathbf{x}_{l_1} \otimes_{\mathrm{Cart}} \mathbf{y}_{l_2})_{l_3} \in (\mathbb{R}^3)^{\otimes l_3}\, \forall\, l_3 \in \{|l_1 - l_2|, \cdots, l_1 + l_2\}$, that is symmetric and traceless. The irreducible Cartesian tensor product is crucial for designing equivariant MPNNs and is used for equivariant convolutions and constructing many-body features in Section 4.2. For an even $l_1 + l_2 - l_3 = 2k$, the general form of an irreducible Cartesian tensor of rank l_3 reads [45]

$$\begin{aligned} &(\mathbf{x}_{l_1} \otimes_{\mathrm{Cart}} \mathbf{y}_{l_2})_{l_3} \\ &\quad = C_{l_1 l_2 l_3} \sum\nolimits_{m=0}^{\min(l_1,l_2)-k} (-1)^m 2^m \frac{(2l_3-2m-1)!!}{(2l_3-1)!!} \big\{ (\mathbf{x}_{l_1} \cdot (k+m) \cdot \mathbf{y}_{l_2}) \otimes \mathbf{I}^{\otimes m} \big\}, \end{aligned} \tag{2}$$

where $(\mathbf{x}_{l_1} \cdot (k+m) \cdot \mathbf{y}_{l_2}) = \sum_{i_1, \cdots, i_{k+m}} (\mathbf{x}_{l_1})_{i_1 \cdots i_{k+m}} (\mathbf{y}_{l_2})_{i_1 \cdots i_{k+m}}$ denotes an $(k+m)$-fold tensor contraction, which results in a tensor of rank $l_1 + l_2 - 2(k+m)$. For simplicity, we skip the uncontracted indices in the above definition. For example, for $l_1 = 4$ and $l_2 = 3$ and a three-fold tensor contraction we obtain $(\mathbf{x}_{l_1} \cdot 3 \cdot \mathbf{y}_{l_2})_{i_4} = \sum_{i_1, i_2, i_3} (\mathbf{x}_{l_1})_{i_1 i_2 i_3 i_4} (\mathbf{y}_{l_2})_{i_1 i_2 i_3}$, i.e., the corresponding tensors are contracted along i_1, i_2, and i_3. Note that the final result is independent of the index permutation, as the contracted tensors are symmetric. For an odd $l_1 + l_2 - l_3 = 2k+1$, we define [45]

$$\begin{aligned} &(\mathbf{x}_{l_1} \otimes_{\mathrm{Cart}} \mathbf{y}_{l_2})_{l_3} \\ &\quad = D_{l_1 l_2 l_3} \sum\nolimits_{m=0}^{\min(l_1,l_2)-k-1} (-1)^m 2^m \frac{(2l_3-2m-1)!!}{(2l_3-1)!!} \big\{ (\boldsymbol{\varepsilon} : \mathbf{x}_{l_1} \cdot (k+m) \cdot \mathbf{y}_{l_2}) \otimes \mathbf{I}^{\otimes m} \big\}, \end{aligned} \tag{3}$$

with $\boldsymbol{\varepsilon}$ denoting the Levi-Civita symbol ($\varepsilon_{i_1 i_2 i_3} = -\varepsilon_{i_3 i_2 i_1}$ and $\varepsilon_{i_1 i_1 i_3} = 0$). The double contraction with the Levi-Civita symbol reads $(\boldsymbol{\varepsilon} : \mathbf{x}_{l_1} \cdot (k+m) \cdot \mathbf{y}_{l_2})_{i_1} = \sum_{i_2, i_3} \varepsilon_{i_1 i_2 i_3} (\mathbf{x}_{l_1} \cdot (k+m) \cdot \mathbf{y}_{l_2})_{i_2 i_3}$, and yields a tensor of rank $l_1 + l_2 - 2(k+m) - 1$. Details on the normalization constants $C_{l_1 l_2 l_3}$ and $D_{l_1 l_2 l_3}$ are provided in Appendix B.1.

4.2 Equivariant message-passing based on irreducible Cartesian tensors

The following section introduces the basic operations for constructing equivariant MPNNs based on irreducible Cartesian tensors. Using their irreducible tensor products, we demonstrate how to build equivariant two- and many-body features, crucial for modeling many-body interactions in molecular and materials systems. Particularly, we focus on MLIPs based on equivariant MPNNs and extend the state-of-the-art MACE architecture [33] to the Cartesian basis. Following the MACE architecture, we use only even tensor products. We split vectors $\mathbf{r}_{uv} = \mathbf{r}_u - \mathbf{r}_v \in \mathbb{R}^3$ from atom u to atom v, schematically shown in Fig. 1 (a), into their radial and angular components (unit vectors), i.e.,

 https://doi.org/10.52202/079017-3940

$r_{uv} = \|\mathbf{r}_{uv}\|_2 \in \mathbb{R}$ and $\hat{\mathbf{r}}_{uv} = \mathbf{r}_{uv}/r_{uv} \in \mathbb{R}^3$, respectively. In the t-th message-passing layer, edges $\{u, v\}$ are embedded using a fully connected neural network $R^{(t)}_{kl_1l_2l_3} : \mathbb{R} \to \mathbb{R}$ with k output feature channels. The radial function $R^{(t)}_{kl_1l_2l_3}$ takes as an input radial distances r_{uv}, which are embedded through Bessel functions and multiplied by a smooth polynomial cutoff function [60]. Finally, we use irreducible Cartesian tensors $\mathbf{T}_l(\hat{\mathbf{r}})$, similar to spherical tensors $\mathbf{Y}^l(\hat{\mathbf{r}})$, to embed unit vectors into the tensor space of maximal rank $l_{\max}$.

Equivariant convolutions and two-body features. Rotation equivariance in MPNNs is typically achieved by constraining convolution filters to be the products between learnable radial functions and spherical tensors, i.e., $R^{(t)}_{kl_1l_2l_3}(r_{uv})Y^{l_1}_{m_1}(\hat{\mathbf{r}}_{uv})$. The two-body features $A^{(t)}_{ukl_3m_3}$ are further obtained through the tensor product—the point-wise convolution [39]—between the respective filters and neighbors' equivariant features $h^{(t)}_{ukl_2m_2}$. The permutational invariance is enforced by pooling over the neighbors $v \in \mathcal{N}(u)$. Here, we use irreducible Cartesian tensors, with the rotation-equivariant filters given by $R^{(t)}_{kl_1l_2l_3}(r_{uv})\big(\mathbf{T}_{l_1}(\hat{\mathbf{r}}_{uv})\big)_{i_1i_2\cdots i_{l_1}}$. Thus, two-body features $\big(\mathbf{A}^{(t)}_{ukl_3}\big)_{i_1i_2\cdots i_{l_3}}$ are obtained using the irreducible Cartesian tensor product and are represented by rank-l_3 irreducible Cartesian tensors. The Cartesian two-body features are defined by

$$\big(\mathbf{A}^{(t)}_{ukl_3}\big)_{i_1i_2\cdots i_{l_3}} = \sum\nolimits_{v\in\mathcal{N}(u)} \Big(R^{(t)}_{kl_1l_2l_3}(r_{uv})\mathbf{T}_{l_1}(\hat{\mathbf{r}}_{uv}) \otimes_{\text{Cart}} \frac{1}{\sqrt{d_t}} \sum_{k'} W^{(t)}_{kk'l_2}\mathbf{h}^{(t)}_{vk'l_2}\Big)_{i_1i_2\cdots i_{l_3}}, \quad (4)$$

where d_t represents the number of feature channels in the node embeddings $\mathbf{h}^{(t)}_{vk'l_2}$ of the t-th message-passing layer. In the first message-passing layer, node embeddings are initialized as learnable weights W_{kZ_v} that are invariant to actions of the orthogonal group, i.e., are scalars or tensors of rank $l_2 = 0$, and embed the atom type Z_v. Thus, constructing equivariant two-body features simplifies to

$$\big(\mathbf{A}^{(1)}_{ukl_1}\big)_{i_1i_2\cdots i_{l_1}} = \sum\nolimits_{v\in\mathcal{N}(u)} R^{(1)}_{kl_1}(r_{uv})\big(\mathbf{T}_{l_1}(\hat{\mathbf{r}}_{uv})\big)_{i_1i_2\cdots i_{l_1}} W_{kZ_v}. \quad (5)$$

Equivariant many-body features. The importance of many-body terms arises from the fact that the interaction between atoms changes when additional atoms are present; see Appendix A. Furthermore, many-body terms are often required to ensure the generalization of interatomic potentials, i.e., their ability to accurately predict energies and forces for temperatures and stoichiometries on which they were not trained [71]. Here, we construct $(\nu+1)$-body equivariant features from $\big(\mathbf{A}^{(t)}_{ukl_\xi}\big)_{i_1i_2\cdots i_{l_\xi}}$ obtained using Eqs. (4) or (5). The ν-fold Cartesian tensor product, which yields $(\nu+1)$-body features represented by an irreducible Cartesian tensor of rank L, reads

$$\big(\mathbf{B}^{(t)}_{u\eta_\nu kL}\big)_{i_1i_2\cdots i_L} = \big(\underbrace{\tilde{\mathbf{A}}^{(t)}_{ukl_1} \otimes_{\text{Cart}} \cdots \otimes_{\text{Cart}} \tilde{\mathbf{A}}^{(t)}_{ukl_\nu}}_{\nu\text{-fold}}\big)_{i_1i_2\cdots i_L}, \quad (6)$$

where η_ν counts all possible ν-fold products of $\{l_1, \cdots, l_\nu\}$-rank tensors, yielding rank-L irreducible Cartesian tensors, and $\tilde{\mathbf{A}}^{(t)}_{ukl_\xi} = \frac{1}{\sqrt{d_t}}\sum_{k'} W^{(t)}_{kk'l_\xi}\mathbf{A}^{(t)}_{uk'l_\xi}$ with d_t feature channels.

The irreducible Cartesian tensor product does not allow pre-computing the coefficients of the ν-fold tensor product, differing from spherical tensors that use the generalized Clebsch–Gordan coefficients, contracted with weights from Eq. (A1) and summed over the possible paths $\text{len}(\eta_\nu)$, for this purpose [33, 40, 51]. Thus, we obtain the result of Eq. (6) by iteratively applying the irreducible Cartesian tensor product $(\nu-1)$ times and refer to the respective models as irreducible Cartesian tensor potentials (ICTPs) with the full product basis or $\text{ICTP}_{\text{full}}$. However, two-fold tensor products in Eq. (6) are symmetric to permutations of involved tensors. Thus, the number of the ν-fold tensor products, $\text{len}(\eta_\nu)$, can be significantly reduced; we refer to the corresponding models as ICTP_{sym}. Furthermore, we can reduce the computational cost of the Cartesian product basis by performing the calculations in a latent feature space. We use learnable weights $W_{kk'l_\xi}$ to reduce the number of feature channels for the product basis calculation and then increase it again for subsequent steps; we refer to the corresponding models as ICTP_{lt}. For more details on the model architecture, such as the construction of updated many-body node embeddings, readout functions, and different options for the Cartesian product basis, see Appendix B.2.

Runtime considerations. When choosing an architecture to implement MLIPs, the runtime per energy and force evaluation is crucial. The computational complexity as a function of rank L is

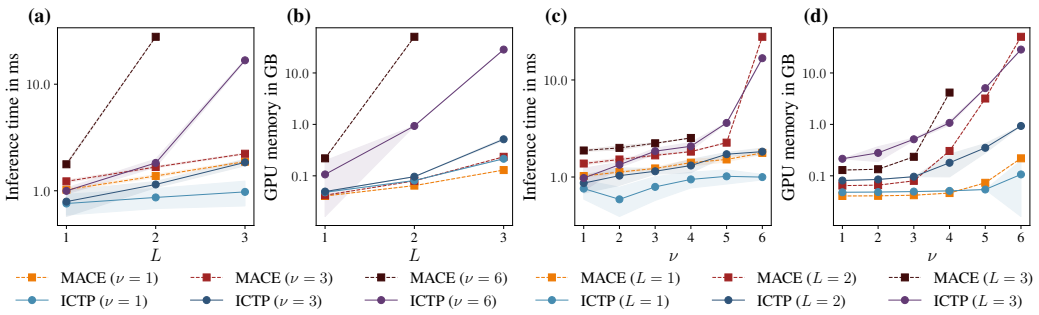

Figure 2: **Inference times and memory consumption as a function of the tensor rank L (a)–(b) and the correlation order ν (c)–(d).** All results are obtained for the 3BPA data set and $l_{\max} = L$. We used eight feature channels to allow experiments with larger ν values. MACE models use intermediate tensors with $l > l_{\max}$ for their product basis, which we fixed to $l = l_{\max}$. Otherwise, pre-computing generalized Clebsch–Gordan coefficients for $\nu > 4$ would require more than 2 TB of RAM. For ICTP, we used the full product basis to compute the same number of ν-fold tensor products as in MACE.

$\mathcal{O}\left(9^L L!/\left(2^{L/2}\,(L/2)!\right)\right)$ for the irreducible Cartesian tensor product and $\mathcal{O}\left(L^5\right)$ for the Clebsch–Gordan one; see Appendix B.3 for more details. Thus, equivariant convolutions based on spherical tensors are more computationally efficient for $L \to \infty$ than those based on irreducible Cartesian tensors. However, state-of-the-art models and physical properties typically require $L \leq 4$ [33, 72], making sub-leading terms and implementation-dependent amplitudes crucial. Based on our analysis, for $L \leq 4$, we can expect equivariant convolutions based on irreducible Cartesian tensors to be more computationally and memory efficient than their spherical counterparts. For $L \leq 4$, the ν-fold tensor product in the Cartesian basis can also offer computational advantages. Its cost, as a function of rank L and correlation order ν, is $\mathcal{K}(9^L L!/(2^{L/2}(L/2)!))^{\nu-1}$ and $\mathcal{K}L^{\frac{1}{2}\nu(\nu+3)}$ for ICTP and MACE, respectively. The pre-factor $\mathcal{K}$, which counts all possible ν-fold tensor products, is removed in MACE through generalized Clebsch-Gordan coefficients, though these coefficients increase the memory spherical models use. Therefore, it is essential to consider the inference times for a fair comparison, which we provide in Section 5.

Equivariance and tracelss property of message-passing layers. We conclude this section by giving a theoretical result that ensures the equivariance of message-passing layers based on irreducible Cartesian tensors and their irreducible tensor products to actions of the orthogonal group. The proof is provided in Appendix C. We also prove in Appendix D that these message-passing layers preserve the traceless property of irreducible Cartesian tensors.

Proposition 4.1. *The message-passing layers based on irreducible Cartesian tensors and their irreducible tensor products are equivariant to actions of the orthogonal group.*

Proposition 4.2. *The message-passing layers based on irreducible Cartesian tensors and their irreducible tensor products preserve the traceless property of irreducible Cartesian tensors.*

5 Experiments and results

This section presents the results for the five benchmark data sets: rMD17, MD22, 3BPA, acetylacetone, and Ta–V–Cr–W. We describe data sets and training details in Appendices E.1 and E.2, respectively.

Scaling and computational cost. The expressive power and computational efficiency of equivariant many-body message-passing potentials depend on the tensor ranks employed in equivariant message passing and embedding local atomic environments, i.e., L and $l_{\max}$, respectively, as well as the correlation order ν. Recent work has shown that for identifying environments with L-fold symmetries, at least rank-L tensors are required [73]. These symmetries are typically lifted in atomistic simulations, motivating the use of $L \leq 4$. Higher body-order correlations ν, in turn, are required if atomic environments are degenerate to a lower body-order correlation $\nu - 1$ [73, 74]. Figure 2 demonstrates inference times and memory consumption of models based on irreducible Cartesian and spherical tensors, i.e., ICTP and MACE, respectively, as a function of the tensor rank and the correlation order. Table A1 presents the corresponding numerical results. We find that irreducible Cartesian tensors

https://doi.org/10.52202/079017-3940

Table 1: **Energy (E) and force (F) mean absolute errors (MAEs) for the rMD17 data set.** E- and F-MAE are given in meV and meV/Å, respectively. Results are shown for models trained using $N_{\text{train}} = \{950, 50\}$ configurations randomly drawn from the data set, with further 50 used for early stopping. All values are obtained by averaging over five independent runs, with the standard deviation provided if available. Best performances, considering the standard deviation, are highlighted in bold.

		$N_{\text{train}} = 950$					$N_{\text{train}} = 50$		
		ICTP$_{\text{sym}}$	TensorNet [30]	MACE [33]	Allegro [34]	NequIP [25]	NequIP [25]	MACE [33]	ICTP$_{\text{sym}}$
Aspirin	E	**2.27 ± 0.11**	2.4	**2.2**	2.3	2.3	19.5	17.0	**14.84 ± 0.98**
	F	**6.67 ± 0.19**	8.9 ± 0.1	**6.6**	7.3	8.2	52.0	43.9	**40.19 ± 0.95**
Azobenzene	E	1.20 ± 0.01	**0.7**	1.2	1.2	**0.7**	6.0	**5.4**	**5.47 ± 0.63**
	F	2.92 ± 0.03	3.1	3.0	**2.6**	2.9	20.0	**17.7**	**17.25 ± 0.53**
Benzene	E	0.26 ± 0.00	**0.02**	0.4	0.3	0.04	0.6	0.7	**0.38 ± 0.02**
	F	0.34 ± 0.02	0.3	0.3	**0.2**	0.3	2.9	2.7	**2.45 ± 0.13**
Ethanol	E	**0.43 ± 0.02**	0.5	**0.4**	**0.4**	**0.4**	8.7	6.7	**6.15 ± 0.26**
	F	2.63 ± 0.10	3.5	**2.1**	**2.1**	2.8	40.2	32.6	**29.53 ± 1.14**
Malonaldehyde	E	0.82 ± 0.03	0.8	0.8	**0.6**	0.8	12.7	**10.0**	**9.72 ± 0.42**
	F	4.96 ± 0.21	5.4	4.1	**3.6**	5.1	52.5	**43.3**	**42.88 ± 3.08**
Naphthalene	E	0.56 ± 0.00	**0.2**	0.5	**0.2**	0.9	**2.1**	**2.1**	**2.06 ± 0.10**
	F	1.45 ± 0.05	1.6	1.6	**0.9**	1.3	10.0	**9.2**	**9.43 ± 0.46**
Paracetamol	E	1.44 ± 0.03	**1.3**	**1.3**	1.5	1.4	14.3	9.7	**8.94 ± 0.66**
	F	**4.89 ± 0.11**	5.9 ± 0.1	**4.8**	4.9	5.9	39.7	**31.5**	**30.13 ± 1.51**
Salicylic acid	E	0.97 ± 0.01	0.8	0.9	0.9	**0.7**	8.0	6.5	**5.95 ± 0.43**
	F	3.66 ± 0.06	4.6 ± 0.1	3.1	**2.9**	4.0	35.9	**28.4**	**27.78 ± 1.93**
Toluene	E	0.46 ± 0.00	**0.3**	0.5	0.4	**0.3**	3.3	3.1	**2.45 ± 0.13**
	F	1.61 ± 0.02	1.7	**1.5**	1.8	1.6	15.1	12.1	**11.24 ± 0.55**
Uracil	E	0.57 ± 0.01	**0.4**	0.5	0.6	**0.4**	7.3	**4.4**	4.66 ± 0.16
	F	2.64 ± 0.08	3.1	2.1	**1.8**	3.1	40.1	**25.9**	**25.97 ± 0.78**

outperform spherical ones for most parameter values. In particular, irreducible Cartesian tensors allow spanning the ν-space more efficiently, in line with our theoretical results in Section 4.2.

Molecular dynamics trajectories. We assess the performance of ICTP models using the revised MD17 (rMD17) data set, which includes structures, total energies, and atomic forces for ten small organic molecules obtained from ab initio molecular dynamics simulations [47]. Table 1 shows that ICTP$_{\text{sym}}$ achieves accuracy on par with state-of-the-art spherical and Cartesian models. Notably, several methods exhibit similar accuracy when trained with 950 configurations. However, the achieved accuracy is much lower than the desired accuracy of 43.37 meV ≈ 1 kcal/mol, making a model comparison less meaningful. Therefore, we also compare ICTP$_{\text{sym}}$ with MACE and NequIP, trained using 50 configurations, making learning accurate MLIPs more challenging. From Table 1, we see that ICTP$_{\text{sym}}$ outperforms MACE and NequIP for most molecules in this scenario.

We further evaluate the performance of ICTP using the MD22 data set, which contains seven molecular systems with 42–370 atoms [48]. This data set spans four major classes of biomolecules and supramolecules and was designed to challenge short-range models. Table A2 shows that ICTP achieves an accuracy on par with or better than state-of-the-art models, including long-range ones.

Extrapolation to out-of-domain data. We continue to assess the performance of ICTP models using the 3BPA data set [49]. The training data set comprises 500 configurations, total energies, and atomic forces acquired from molecular dynamics at 300 K. The test data set is obtained from simulations at 300 K, 600 K, and 1200 K. We also test our models using energies and forces along dihedral rotations of the molecule. Table 2 shows that ICTP models trained using 450 configurations perform on par with state-of-the-art spherical models, similar to the results for rMD17. However, we were not able to reproduce the original results using the current MACE source code and the described training setup [33]. Therefore, for a fair comparison, we unified the training setup for ICTP and MACE (see Appendix E.2) and Table 2 reports the corresponding results for 450 training configurations. In Table A3, we present the results obtained for ICTP and MACE trained using 50 configurations.

From Table 2, we observe that ICTP$_{\text{full}}$ slightly outperforms MACE in total energy and atomic force RMSEs but is ~ 1.4 times less computationally efficient. This difference arises from MACE using the generalized Clebsch–Gordan coefficients and pre-computing their product with the weights in the linear expansion in Eq. (A1) [33], which reduces the effective number of evaluated tensor products. Thus, we may attribute the lower computational efficiency of ICTP$_{\text{full}}$ to the use of the

Table 2: **Energy (E) and force (F) root-mean-square errors (RMSEs) for the 3BPA data set.** E- and F-RMSE are given in meV and meV/Å, respectively. Results are shown for models trained using 450 configurations randomly drawn from the training data set collected at 300 K, with further 50 used for early stopping. All ICTP results are obtained by averaging over five independent runs. For MACE and NequIP, the results are reported for three runs. The standard deviation is provided if it is available. Best performances, considering the standard deviation, are highlighted in bold. Inference time and memory consumption are measured for a batch size of 100. Inference time is reported per structure in ms, while memory consumption is provided for the entire batch in GB.

		$\text{ICTP}_{\text{full}}$	ICTP_{sym}	$\text{ICTP}_{\text{sym+lt}}$	MACE[a]	CACE [31]	MACE [33]	NequIP [34]
300 K	E	**2.70 ± 0.22**	**2.70 ± 0.08**	**2.98 ± 0.34**	**2.81 ± 0.18**	6.3	**3.0 ± 0.2**	3.28 ± 0.10
	F	**9.45 ± 0.29**	**9.39 ± 0.31**	**9.57 ± 0.20**	**9.47 ± 0.42**	21.4	**8.8 ± 0.3**	10.77 ± 0.19
600 K	E	**10.74 ± 0.31**	**10.38 ± 0.80**	**10.29 ± 0.90**	**11.11 ± 1.41**	18.0	**9.7 ± 0.5**	11.16 ± 0.14
	F	**22.99 ± 0.64**	**22.87 ± 0.91**	**23.03 ± 0.76**	**23.27 ± 1.45**	45.2	**21.8 ± 0.6**	26.37 ± 0.09
1200 K	E	**29.80 ± 0.92**	**30.84 ± 1.87**	**31.32 ± 1.80**	**31.15 ± 1.58**	58.0	**29.8 ± 1.0**	38.52 ± 1.63
	F	**62.82 ± 1.23**	**64.54 ± 3.88**	**65.36 ± 3.47**	**65.22 ± 3.52**	113.8	**62.0 ± 0.7**	76.18 ± 1.11
Dihedral slices	E	**9.82 ± 0.79**	10.64 ± 1.07	13.03 ± 3.44	**8.56 ± 1.53**	–	**7.8 ± 0.6**	23.2 [33]
	F	**17.52 ± 0.54**	**17.18 ± 0.81**	19.31 ± 0.83	**17.69 ± 1.29**	–	**16.5 ± 1.7**	23.1 [33]
Inference time		6.45 ± 0.50	5.31 ± 0.02	**3.51 ± 0.22**	4.66 ± 0.05	–	**24.3**[b]	103.5[b] [33]
Memory consumption		49.66 ± 0.00	42.01 ± 0.11	39.08 ± 0.00	**36.26 ± 0.00**	–	–	–

[a] During inference time measurements with the MACE source code, we were not able to reproduce the original results [33]. Thus, we re-run MACE experiments using a training setup similar to that of ICTP; see Appendix E.2.
[b] The original publication did not report the batch size used to measure inference time [33]. Therefore, the values provided are used solely to demonstrate the relative computational cost of MACE and NequIP.

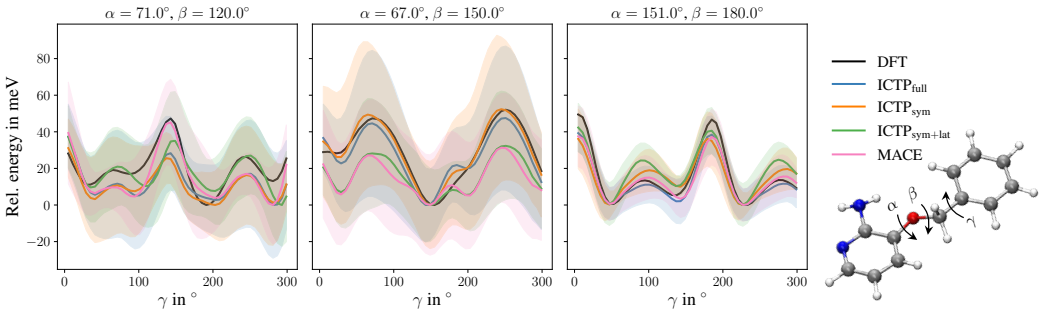

Figure 3: **Potential energy profiles for three cuts through the 3BPA molecule's potential energy surface.** All models are trained using 50 configurations, and additional 50 are used for early stopping. The 3BPA molecule, including the three dihedral angles (α, β, and γ), provided in degrees °, is shown as an inset. The color code of the inset molecule is C grey, O red, N blue, and H white. The reference potential energy profile (DFT) is shown in black. Each profile is shifted such that each model's lowest energy is zero. Shaded areas denote standard deviations across five independent runs.

MACE architecture, which results in a larger pre-factor $\mathcal{K}$ for Cartesian models but facilitates a fair comparison between irreducible Cartesian and spherical tensors. Using the symmetric Cartesian product basis and that in the latent space, for example, we further improve the runtime of our models while maintaining accuracy on par with MACE.

Table A4 presents additional results obtained for $\nu = 1$, i.e., focusing on models that rely exclusively on two-body interactions. We observe that Cartesian models exhibit shorter inference times than spherical ones, with MACE and $\text{ICTP}_{\text{full}}$ achieving 2.96 ± 0.06 ms and 2.63 ± 0.02 ms, respectively. Regarding memory consumption, MACE and ICTP perform similarly despite the larger number of tensor products required for the Cartesian product basis in Eq. (6). This observation can be attributed to, for example, the Clebsch–Gordan tensor product requiring the computation of intermediate tensors with $(2L+1)^2$ elements, whereas irreducible Cartesian tensors contain 3^L elements.

Figure 3 compares potential energy profiles obtained with ICTP and MACE trained using 50 configurations. For the potential energy cut at $\beta = 120°$ (left panel), ICTP and MACE perform similarly, except for the energy barrier at $\gamma \approx 143°$, which ICTP tends to underestimate stronger than MACE.

https://doi.org/10.52202/079017-3940

Table 3: **Energy (E) and force (F) root-mean-square errors (RMSEs) for the acetylacetone data set.** E- and F-RMSE are given in meV and meV/Å, respectively. Results are shown for models trained using 450 configurations randomly drawn from the training data set collected at 300 K, with further 50 used for early stopping. All ICTP results are obtained by averaging over five independent runs. For MACE and NequIP, the results are reported for three runs. The standard deviation is provided if it is available. Best performances, considering the standard deviation, are highlighted in bold.

		$\text{ICTP}_{\text{full}}$	ICTP_{sym}	$\text{ICTP}_{\text{sym+lt}}$	MACE[a]	MACE [33]	NequIP [32]
300 K	E	**0.75 ± 0.04**	**0.76 ± 0.03**	**0.77 ± 0.04**	**0.75 ± 0.05**	0.9 ± 0.03	**0.81 ± 0.05**
	F	**5.08 ± 0.11**	**5.17 ± 0.10**	**5.18 ± 0.16**	**5.00 ± 0.17**	**5.1 ± 0.1**	5.90 ± 0.46
600 K	E	**5.39 ± 1.22**	**4.43 ± 0.34**	**5.12 ± 0.29**	**4.96 ± 0.64**	**4.6 ± 0.3**	6.04 ± 1.54
	F	**23.21 ± 1.96**	**22.90 ± 1.62**	**24.05 ± 1.71**	**23.25 ± 1.82**	**22.4 ± 0.9**	27.80 ± 4.03
Number of parameters		2,774,800	2,736,400	2,648,080	2,803,984	2,803,984	3,190,488

[a] Similar to Table 2, we re-run MACE experiments using the similar training setup as for ICTP; see Appendix E.2.

For $\beta = 150°$ (middle panel), however, $\text{ICTP}_{\text{full}}$ and ICTP_{sym} outperform MACE across nearly the entire range of the dihedral angle γ. For $\beta = 180°$ (right panel), all models perform similarly. Figure A1 shows the corresponding potential energy profiles for models trained with 450 configurations. All models perform similarly in this scenario, with energy profiles close to the reference (DFT).

Flexibility and reactivity. We further use the acetylacetone data set to assess the ICTP models' extrapolation capabilities to higher temperatures (similar to 3BPA), bond breaking, and bond torsions [32]. Table 3 shows that ICTP models achieve state-of-the-art results while employing fewer parameters than spherical counterparts. Appendix E.3 includes additional results for the acetylacetone data set, such as total energy and atomic force RMSEs for models trained with 50 configurations and details on the potential energy profiles for hydrogen transfer and C-C bond rotation. Overall, ICTP and MACE perform similarly, demonstrating excellent generalization capability. However, when trained using 50 configurations, $\text{ICTP}_{\text{full}}$ is the only MLIP consistently producing the potential energy profile for hydrogen transfer close to the reference (DFT).

Multicomponent alloys. We finally evaluate the ICTP and MACE models using the Ta–V–Cr–W data set, designed to assess the performance of state-of-the-art MLIPs in modeling chemically complex multicomponent systems. In this evaluation, we attempt to predict energies and forces for Ta–V–Cr–W subsystems under two scenarios: The 0 K energies and forces in binary, ternary, and quaternary systems and near-melting temperature energies and forces in 4-component disordered alloys. Table A6 shows that ICTP_{sym} outperforms MACE in nearly all subsystems, particularly in energy prediction. ICTP achieves an overall accuracy of 1.38 ± 0.09 meV/atom for energies and 0.028 ± 0.001 eV/Å for forces, compared to 2.19 ± 0.31 meV/atom and 0.029 ± 0.001 eV/Å by MACE. However, the $\mathcal{K}$ pre-factor from the ν-fold tensor product results in longer inference times for ICTP than MACE, in line with the discussion for 3BPA.

6 Conclusions and limitations

This work introduces many-body equivariant MPNNs based on higher-rank irreducible Cartesian tensors, offering an alternative to spherical models and addressing the limitations of state-of-the-art Cartesian models. We assess the performance of resulting MPNNs using five benchmark data sets, such as rMD17, MD22, 3BPA, acetylacetone, and Ta–V–Cr–W. In these experiments, MPNNs based on irreducible Cartesian tensors show a lower computational cost of individual operations compared to spherical counterparts. Furthermore, we demonstrate that these Cartesian models achieve accuracy and generalization capability on par with or better than state-of-the-art spherical models while memory consumption is comparable. Our results hold across the typical range of tensor ranks used in modeling many-body interactions and relevant physical properties, i.e., $L \leq 4$.

Limitations. We emphasize our focus on introducing MPNNs based on irreducible Cartesian tensors and prove their equivariance and traceless property. We adapted the MACE architecture, which uses only even tensor products, to enable a fair comparison with state-of-the-art spherical models. Further modifications to the architecture are possible and necessary, e.g., to reduce the pre-factor arising from the Cartesian product basis, before we can fully exploit the potential of irreducible Cartesian tensors.

Data availability

All data sets used in this study are publicly available: rMD17 (https://doi.org/10.6084/m9.figshare.12672038.v3), MD22 (http://www.sgdml.org), 3BPA (https://github.com/davkovacs/BOTNet-datasets), acetylacetone (https://github.com/davkovacs/BOTNet-datasets), and Ta–V–Cr–W (https://doi.org/10.18419/darus-3516).

Code availability

The source code is available on GitHub and can be accessed via this link: https://github.com/nec-research/ictp.

Acknowledgements

MN acknowledges support from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2075 – 390740016 and the Stuttgart Center for Simulation Science (SimTech).

References

[1] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh: *Machine learning for molecular and materials science*. Nature **559**, 547–555 (2018)

[2] J. Vamathevan, D. Clark, P. Czodrowski, I. Dunham, E. Ferran et al.: *Applications of machine learning in drug discovery and development*. Nat. Rev. Drug Discov. **18**, 463–477 (2019)

[3] J. A. Keith, V. Vassilev-Galindo, B. Cheng, S. Chmiela, M. Gastegger et al.: *Combining Machine Learning and Computational Chemistry for Predictive Insights Into Chemical Systems*. Chem. Rev. **121**, 9816–9872 (2021)

[4] O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky et al.: *Machine Learning Force Fields*. Chem. Rev. **121**, 10142–10186 (2021)

[5] N. Fedik, R. Zubatyuk, M. Kulichenko, N. Lubbers, J. S. Smith et al.: *Extending machine learning beyond interatomic potentials for predicting molecular properties*. Nat. Rev. Chem. **6**, 653–672 (2022)

[6] A. Merchant, S. Batzner, S. S. Schoenholz, M. Aykol, G. Cheon et al.: *Scaling deep learning for materials discovery*. Nature **624**, 80–85 (2023)

[7] D. P. Kovács, J. H. Moore, N. J. Browning, I. Batatia, J. T. Horton et al.: *MACE-OFF23: Transferable Machine Learning Force Fields for Organic Molecules*. https://arxiv.org/abs/2312.15211 (2023)

[8] I. Batatia, P. Benner, Y. Chiang, A. M. Elena, D. P. Kovács et al.: *A foundation model for atomistic materials chemistry*. https://arxiv.org/abs/2401.00096 (2023)

[9] G. D. Purvis and R. J. Bartlett: *A full coupled-cluster singles and doubles model: The inclusion of disconnected triples*. J. Chem. Phys. **76**, 1910–1918 (1982)

[10] T. D. Crawford and H. F. Schaefer III: *An Introduction to Coupled Cluster Theory for Computational Chemists*, pp. 33–136. John Wiley & Sons, Ltd (2000)

[11] R. J. Bartlett and M. Musiał: *Coupled-cluster theory in quantum chemistry*. Rev. Mod. Phys. **79**, 291–352 (2007)

[12] P. Hohenberg and W. Kohn: *Inhomogeneous Electron Gas*. Phys. Rev. **136**, B864–B871 (1964)

[13] W. Kohn and L. J. Sham: *Self-Consistent Equations Including Exchange and Correlation Effects*. Phys. Rev. **140**, A1133–A1138 (1965)

[14] A. Micheli: *Neural network for graphs: A contextual constructive approach*. IEEE Trans. Neural Netw. **20**, 498–511 (2009)

[15] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini: *The graph neural network model*. IEEE Trans. Neural Netw. **20**, 61–80 (2009)

[16] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl: *Neural Message Passing for Quantum Chemistry*. Int. Conf. Mach. Learn. **70**, 1263–1272 (2017)

[17] W. L. Hamilton, R. Ying, and J. Leskovec: *Representation learning on graphs: Methods and applications*. IEEE Data Eng. Bull. **40**, 52–74 (2017)

[18] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst: *Geometric deep learning: going beyond Euclidean data*. IEEE Signal Process. Mag. **34**, 18–42 (2017)

[19] Z. Zhang, P. Cui, and W. Zhu: *Deep Learning on Graphs: A Survey*. IEEE Trans. Knowl. Data Eng. **34**, 249–270 (2022)

[20] S. Zhang, H. Tong, J. Xu, and R. Maciejewski: *Graph convolutional networks: a comprehensive review*. Comput. Soc. Netw. **6**, 11 (2019)

[21] D. Bacciu, F. Errica, A. Micheli, and M. Podda: *A Gentle Introduction to Deep Learning for Graphs*. Neural Netw. **129**, 203–221 (2020)

[22] C. L. Zitnick, L. Chanussot, A. Das, S. Goyal, J. Heras-Domingo et al.: *An Introduction to Electrocatalyst Design using Machine Learning for Renewable Energy Storage*. `https://arxiv.org/abs/2010.09435` (2020)

[23] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov et al.: *Highly accurate protein structure prediction with AlphaFold*. Nature **596**, 583–589 (2021)

[24] J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte et al.: *Robust deep learning–based protein sequence design using ProteinMPNN*. Science **378**, 49–56 (2022)

[25] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa et al.: *E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials*. Nat. Commun. **13**, 2453 (2022)

[26] A. Duval, S. V. Mathis, C. K. Joshi, V. Schmidt, S. Miret et al.: *A Hitchhiker's Guide to Geometric GNNs for 3D Atomic Systems*. `https://arxiv.org/abs/2312.07511` (2024)

[27] M. F. Langer, A. Goeßmann, and M. Rupp: *Representations of molecules and materials for interpolation of quantum-mechanical simulations via machine learning*. npj Comput. Mater. **8**, 41 (2022)

[28] K. T. Schütt, O. T. Unke, and M. Gastegger: *Equivariant message passing for the prediction of tensorial properties and molecular spectra*. Int. Conf. Mach. Learn. **139**, 9377–9388 (2021)

[29] M. Haghighatlari, J. Li, X. Guan, O. Zhang, A. Das et al.: *NewtonNet: a Newtonian message passing network for deep learning of interatomic potentials and forces*. Digital Discovery **1**, 333–343 (2022)

[30] G. Simeon and G. D. Fabritiis: *TensorNet: Cartesian Tensor Representations for Efficient Learning of Molecular Potentials*. Adv. Neural Inf. Process. Syst. **36**, 37334–37353 (2023)

[31] B. Cheng: *Cartesian atomic cluster expansion for machine learning interatomic potentials*. npj Comput. Mater. **10**, 157 (2024)

[32] I. Batatia, S. Batzner, D. P. Kovács, A. Musaelian, G. N. C. Simm et al.: *The Design Space of E(3)-Equivariant Atom-Centered Interatomic Potentials*. `https://arxiv.org/abs/2205.06643` (2022)

[33] I. Batatia, D. P. Kovacs, G. N. C. Simm, C. Ortner, and G. Csanyi: *MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields*. Adv. Neural Inf. Process. Syst. **35**, 11423–11436 (2022)

[34] A. Musaelian, S. Batzner, A. Johansson, L. Sun, C. J. Owen et al.: *Learning local equivariant representations for large-scale atomistic dynamics*. Nat. Commun. **14**, 579 (2023)

[35] S. Passaro and C. L. Zitnick: *Reducing SO(3) Convolutions to SO(2) for Efficient Equivariant GNNs*. Int. Conf. Mach. Learn. **202**, 27420–27438 (2023)

[36] S. Luo, T. Chen, and A. S. Krishnapriyan: *Enabling Efficient Equivariant Operations in the Fourier Basis via Gaunt Tensor Products*. Int. Conf. Learn. Represent. `https://arxiv.org/abs/2401.10216` (2024)

[37] P. Thölke and G. D. Fabritiis: *Equivariant Transformers for Neural Network based Molecular Potentials*. Int. Conf. Learn. Represent. `https://arxiv.org/abs/2202.02541` (2022)

[38] R. F. Snider: *Irreducible Cartesian Tensors*. De Gruyter, Berlin, Boston (2018)

[39] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li et al.: *Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds*. `https://arxiv.org/abs/1802.08219` (2018)

[40] R. Drautz: *Atomic cluster expansion for accurate and transferable interatomic potentials*. Phys. Rev. B **99**, 014104 (2019)

[41] E. Wigner: *Group Theory and its Application to the Quantum Mechanics of Atomic Spectra*, volume 5. Elsevier (2012)

[42] J. A. R. Coope, R. F. Snider, and F. R. McCourt: *Irreducible Cartesian Tensors*. J. Chem. Phys. **43**, 2269–2275 (1965)

[43] J. A. R. Coope and R. F. Snider: *Irreducible Cartesian Tensors. II. General Formulation*. J. Math. Phys. **11**, 1003–1017 (1970)

[44] J. A. R. Coope: *Irreducible Cartesian Tensors. III. Clebsch-Gordan Reduction*. J. Math. Phys. **11**, 1591–1612 (1970)

[45] D. R. Lehman and W. C. Parke: *Angular reduction in multiparticle matrix elements*. J. Math. Phys. **30**, 2797–2806 (1989)

[46] M. Weiler, M. Geiger, M. Welling, W. Boomsma, and T. S. Cohen: *3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data*. Adv. Neural Inf. Process. Syst. **31** (2018)

[47] A. S. Christensen and O. A. von Lilienfeld: *On the role of gradients for machine learning of molecular energies and forces*. Mach. Learn.: Sci. Technol. **1**, 045018 (2020)

[48] S. Chmiela, V. Vassilev-Galindo, O. T. Unke, A. Kabylda, H. E. Sauceda et al.: *Accurate global machine learning force fields for molecules with hundreds of atoms*. Sci. Adv. **9**, eadf0873 (2023)

[49] D. P. Kovács, C. v. d. Oord, J. Kucera, A. E. A. Allen, D. J. Cole et al.: *Linear Atomic Cluster Expansion Force Fields for Organic Molecules: Beyond RMSE*. J. Chem. Theory Comput. **17**, 7696–7711 (2021)

[50] K. Gubaev, V. Zaverkin, P. Srinivasan, A. I. Duff, J. Kästner et al.: *Performance of two complementary machine-learned potentials in modelling chemically complex systems*. npj Comput. Mater. **9**, 129 (2023)

[51] R. Drautz: *Atomic cluster expansion of scalar, vectorial, and tensorial properties including magnetism and charge transfer*. Phys. Rev. B **102**, 024104 (2020)

[52] B. Anderson, T. S. Hy, and R. Kondor: *Cormorant: Covariant Molecular Neural Networks*. Adv. Neural Inf. Process. Syst. **32**, 14537–14546 (2019)

[53] V. G. Satorras, E. Hoogeboom, and M. Welling: *E(n) Equivariant Graph Neural Networks*. Int. Conf. Mach. Learn. **139**, 9323–9332 (2021)

 https://doi.org/10.52202/079017-3940

[54] J. Brandstetter, R. Hesselink, E. van der Pol, E. J. Bekkers, and M. Welling: *Geometric and Physical Quantities improve E(3) Equivariant Message Passing*. Int. Conf. Learn. Represent. https://arxiv.org/abs/2110.02905 (2022)

[55] K. Schütt, P.-J. Kindermans, H. E. Sauceda Felix, S. Chmiela, A. Tkatchenko et al.: *SchNet: A continuous-filter convolutional neural network for modeling quantum interactions*. Adv. Neural Inf. Process. Syst. **30**, 991–1001 (2017)

[56] O. T. Unke and M. Meuwly: *PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges*. J. Chem. Theory Comput. **15** (6), 3678–3693 (2019)

[57] Y. Liu, L. Wang, M. Liu, Y. Lin, X. Zhang et al.: *Spherical Message Passing for 3D Molecular Graphs*. Int. Conf. Learn. Represent. https://arxiv.org/abs/2102.05013 (2022)

[58] J. Gasteiger, S. Giri, J. T. Margraf, and S. Günnemann: *Fast and Uncertainty-Aware Directional Message Passing for Non-Equilibrium Molecules*. Machine Learning for Molecules Workshop, Adv. Neural Inf. Process. Syst. https://arxiv.org/abs/2011.14115 (2020)

[59] J. Gasteiger, F. Becker, and S. Günnemann: *GemNet: Universal Directional Graph Neural Networks for Molecules*. Adv. Neural Inf. Process. Syst. **34**, 6790–6802 (2021)

[60] J. Gasteiger, J. Groß, and S. Günnemann: *Directional Message Passing for Molecular Graphs*. Int. Conf. Learn. Represent. https://arxiv.org/abs/2003.03123 (2020)

[61] D. P. Kovács, I. Batatia, E. S. Arany, and G. Csányi: *Evaluation of the MACE force field architecture: From medicinal chemistry to materials science*. J. Chem. Phys. **159**, 044118 (2023)

[62] F. Fuchs, D. Worrall, V. Fischer, and M. Welling: *SE(3)-Transformers: 3D Roto-Translation Equivariant Attention Networks*. Adv. Neural Inf. Process. Syst. **33**, 1970–1981 (2020)

[63] T. Frank, O. Unke, and K.-R. Müller: *So3krates: Equivariant attention for interactions on arbitrary length-scales in molecular systems*. Adv. Neural Inf. Process. Syst. **35**, 29400–29413 (2022)

[64] Y.-L. Liao, B. Wood, A. Das, and T. Smidt: *EquiformerV2: Improved Equivariant Transformer for Scaling to Higher-Degree Representations*. Int. Conf. Learn. Represent. https://arxiv.org/abs/2306.12059 (2023)

[65] A. V. Shapeev: *Moment Tensor Potentials: A Class of Systematically Improvable Interatomic Potentials*. Multiscale Model. Simul. **14**, 1153–1173 (2016)

[66] V. Zaverkin and J. Kästner: *Gaussian Moments as Physically Inspired Molecular Descriptors for Accurate and Scalable Machine Learning Potentials*. J. Chem. Theory Comput. **16**, 5410–5421 (2020)

[67] V. Zaverkin, D. Holzmüller, I. Steinwart, and J. Kästner: *Fast and Sample-Efficient Interatomic Neural Network Potentials for Molecules and Materials Based on Gaussian Moments*. J. Chem. Theory Comput. **17**, 6658–6670 (2021)

[68] J. Behler and M. Parrinello: *Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces*. Phys. Rev. Lett. **98**, 146401 (2007)

[69] U. Fano and G. Racah: *Irreducible Tensorial Sets*. Academic Press Inc., New York (1959)

[70] I. M. Gel'fand, R. A. Minlos, and Z. Y. Shapiro: *Representations of the Rotation and Lorentz Groups and Their Applications*. Pergamon Press, Inc. (1963)

[71] S. V. S. Martin H. Müser and L. Pastewka: *Interatomic potentials: achievements and challenges*. Adv. Phys. X **8**, 2093129 (2023)

[72] I. Grega, I. Batatia, G. Csanyi, S. Karlapati, and V. Deshpande: *Energy-conserving equivariant GNN for elasticity of lattice architected metamaterials*. Int. Conf. Learn. Represent. https://arxiv.org/abs/2401.16914 (2024)

[73] C. K. Joshi, C. Bodnar, S. V. Mathis, T. Cohen, and P. Lio: *On the Expressive Power of Geometric Graph Neural Networks*. Int. Conf. Learn. Represent. `https://arxiv.org/abs/2301.09308` (2023)

[74] S. N. Pozdnyakov, M. J. Willatt, A. P. Bartók, C. Ortner, G. Csányi et al.: *Incompleteness of Atomic Structure Representations*. Phys. Rev. Lett. **125**, 166001 (2020)

[75] J. Behler: *Atom-centered symmetry functions for constructing high-dimensional neural network potentials*. J. Chem. Phys. **134**, 074106 (2011)

[76] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi: *Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons*. Phys. Rev. Lett. **104**, 136403 (2010)

[77] A. P. Bartók, R. Kondor, and G. Csányi: *On representing chemical environments*. Phys. Rev. B **87**, 184115 (2013)

[78] M. S. Daw and M. I. Baskes: *Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals*. Phys. Rev. B **29**, 6443–6453 (1984)

[79] Y.-M. Kim, B.-J. Lee, and M. I. Baskes: *Modified embedded-atom method interatomic potentials for Ti and Zr*. Phys. Rev. B **74**, 014101–014112 (2006)

[80] A. Stone: *Transformation between cartesian and spherical tensors*. Mol. Phys. **29**, 1461–1471 (1975)

[81] A. J. Stone: *Properties of Cartesian-spherical transformation coefficients*. J. Phys. A **9**, 485 (1976)

[82] J. M. Normand and J. Raynal: *Relations between Cartesian and spherical components of irreducible Cartesian tensors*. J. Phys. A **15**, 1437 (1982)

[83] K. He, X. Zhang, S. Ren, and J. Sun: *Deep Residual Learning for Image Recognition*. IEEE Conf. Comput. Vis. Pattern Recognit. `https://doi.org/10.1109/CVPR.2016.90` (2016)

[84] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt et al.: *Machine learning of accurate energy-conserving molecular force fields*. Sci. Adv. **3**, e1603015 (2017)

[85] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko: *Quantum-chemical insights from deep tensor neural networks*. Nat. Commun. **8**, 13890 (2017)

[86] S. Chmiela, H. E. Sauceda, K.-R. Müller, and A. Tkatchenko: *Towards exact molecular dynamics simulations with machine-learned force fields*. Nat. Commun. **9**, 3887 (2018)

[87] S. Elfwing, E. Uchibe, and K. Doya: *Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning*. Neural Netw. **107**, 3–11 (2018)

[88] P. Ramachandran, B. Zoph, and Q. V. Le: *Searching for activation functions*. Int. Conf. Learn. Represent. `https://arxiv.org/abs/1710.05941` (2018)

[89] S. J. Reddi, S. Kale, and S. Kumar: *On the Convergence of Adam and Beyond*. Int. Conf. Learn. Represent. `https://arxiv.org/abs/1904.09237` (2018)

[90] Y. Li, Y. Wang, L. Huang, H. Yang, X. Wei et al.: *Long-Short-Range Message-Passing: A Physics-Informed Framework to Capture Non-Local Interaction for Scalable Molecular Dynamics Simulation*. Int. Conf. Learn. Represent. `https://arxiv.org/abs/2304.13542` (2024)

[91] J. T. Frank, O. T. Unke, K.-R. Müller, and S. Chmiela: *A Euclidean transformer for fast and stable machine learned force fields*. Nat. Commun. **15**, 6539 (2024)

Appendices

Appendix Contents

A Background **17**

B Methods **18**

B.1 Normalization constants 18

B.2 Further details on the equivariant message passing 18

B.3 Computational complexity and runtime analysis 19

C Proof of Cartesian message-passing equivariance to actions of the orthogonal group **20**

D Proof of the traceless property for irreducible Cartesian tensors **23**

E Experiments and results **29**

E.1 Description of the data sets 29

E.2 Training details 30

E.3 Additional results 31

F Broader social impact **36**

A Background

Message-passing neural networks. Message-passing neural networks (MPNNs) learn node representations in a graph by iteratively processing local information sent by the nodes' neighbors. The initial features of node u are represented as the vector $\mathbf{x}_u$, and undirected edges $\{u, v\}$ connect pairs of nodes u, v. A node v belongs to the neighborhood of node u, denoted as $\mathcal{N}(u)$, if there exists an edge $\{u, v\}$ in the graph. Typically, the $(t+1)$-th message-passing layer computes a new node u's representation $\mathbf{h}_u^{(t+1)}$ by applying a permutation invariant aggregation function over the neighbors [16, 21]

$$\mathbf{h}_u^{(t+1)} = \phi^{(t)}\Big(\mathbf{h}_u^{(t)}, \sum\nolimits_{v\in\mathcal{N}(u)} \psi^{(t)}\big(\mathbf{h}_u^{(t)}, \mathbf{h}_v^{(t)}\big)\Big),$$

where $\phi^{(t)}, \psi^{(t)}$ are often implemented as learnable fully-connected neural networks (NNs), and $\mathbf{h}_u^{(0)} = \mathbf{x}_u$. To learn a mapping from a learned representation $\mathbf{h}_u^{(t)}$ to the atoms' energies, we can couple T message-passing layers with corresponding readout functions $\mathcal{R}_t$, $t \in \{1, \ldots, T\}$ such that $E_u = \sum_{t=1}^{T} \mathcal{R}_t\big(\mathbf{h}_u^{(t)}\big)$.

Many-body interatomic potentials. Interatomic potentials approximate the potential energy of atoms—the electronic ground state energy—as a function of their coordinates [71]. Many-body potentials naturally arise because the interaction between two atoms is influenced by the presence of additional atoms, changing their electronic structure. This concept is formalized by expanding the atomic energy E_u of a many-atom system into a series of two-body, three-body, and higher-body-order contributions

$$E_u = E^{(1)}(\mathbf{r}_u) + \sum_{v_1} E^{(2)}(\mathbf{r}_u, \mathbf{r}_{v_1}) + \sum_{v_1<v_2} E^{(3)}(\mathbf{r}_u, \mathbf{r}_{v_1}, \mathbf{r}_{v_2}) + \cdots,$$

where $\mathbf{r}_u$ represents the position of atom u and the superscript ν in $E^{(\nu)}$ indicates the order of the many-body interaction. In the absence of external fields, $E^{(\nu)}$ contributions to the atomic energy are invariant to rotations, and the two-body potential depends only on distances $r_{uv} = \|\mathbf{r}_u - \mathbf{r}_v\|_2$

$$E_u = \sum_{v_1} E^{(2)}(r_{uv_1}) + \sum_{v_1<v_2} E^{(3)}(r_{uv_1}, r_{uv_2}, r_{v_1v_2}) + \cdots.$$

Expansions of this form have found a broad application in constructing machine-learned interatomic potentials (MLIPs), significantly advancing the field [40, 65].

Higher-body-order local descriptors. Recent advances in MLIPs have been influenced by moment tensor potentials (MTPs) [65] and atomic cluster expansion (ACE) [40]. These approaches enable systematic construction of higher-body-order polynomial basis functions, encompassing representations like atom-centered symmetry functions (ACSFs) [68, 75], smooth overlap of atomic positions (SOAP) [76, 77], Gaussian moments [66, 67], and embedded atom/multi-scale embedded atom method (EAM/MEAM) potentials [78, 79]. Furthermore, a reducible Cartesian tensor can be represented as a linear combination of irreducible spherical counterparts, and vice versa [40, 51]. More general expressions, including tensor contractions, have also been provided, hinting at the relationship between Cartesian and spherical models [80–82]. Despite the success of MTP and ACE, defining smaller cutoff radii and rigid architecture can result in limited accuracy compared to MPNNs.

Many-body message passing. Designing accurate and computationally efficient interatomic potential models for interacting many-body systems requires including higher-body-order energy contributions and, thus, higher-body-order learnable features. Recently, a new message construction mechanism has been proposed by expanding the messages $\mathbf{m}_u^{(t)}$ to include many-body contributions [33]

$$\mathbf{m}_u^{(t)} = \sum_{v_1} \psi_2^{(t)}\big(\mathbf{h}_u^{(t)}, \mathbf{h}_{v_1}^{(t)}\big) + \sum_{v_1,v_2} \psi_3^{(t)}\big(\mathbf{h}_u^{(t)}, \mathbf{h}_{v_1}^{(t)}, \mathbf{h}_{v_2}^{(t)}\big) + \cdots + \sum_{v_1,\cdots,v_\nu} \psi_{\nu+1}^{(t)}\big(\mathbf{h}_u^{(t)}, \mathbf{h}_{v_1}^{(t)}, \cdots, \mathbf{h}_{v_\nu}^{(t)}\big),$$

where $(\nu+1)$ denotes the order of many-body interactions, defining the number of contracted tensors. Using $\sum_{v_1,\cdots,v_\nu}$ instead of $\sum_{v_1<\cdots<v_\nu}$ circumvents the exponential increase in computational cost with ν. It allows exploiting the product structure of many-body features, which differs from other approaches [58–60].

 https://doi.org/10.52202/079017-3940

B Methods

B.1 Normalization constants

The following provides the normalization constants $C_{l_1 l_2 l_3}$ and $D_{l_1 l_2 l_3}$ for even and odd irreducible Cartesian tensor products in Eqs. (2) and (3), respectively. The respective normalization constants read [45]

$$C_{l_1 l_2 l_3} = \frac{l_1! l_2! (2l_3 - 1)!! ((L_1 + 1)/2)! ((L_2 + 1)/2)!}{l_3! L_1!! L_2!! L_3!! (L/2)!},$$
$$D_{l_1 l_2 l_3} = \frac{2 l_1! l_2! (2l_3 - 1)!! (L_1/2)! (L_2/2)!}{(l_3 - 1)! (L_1 + 1)!! (L_2 + 1)!! (L_3 + 1)!! ((L + 1)/2)!},$$

with $L = l_1 + l_2 + l_3$ and $L_i = L - 2l_i - 1$. Here, $C_{l_1 l_2 l_3}$ is defined such that an l_3-fold contraction of the tensor $\mathbf{z}_{l_3}$, obtained through the irreducible Cartesian tensor product between $\mathbf{x}_{l_1}$ and $\mathbf{y}_{l_2}$ ($\mathbf{x}_{l_1}$ and $\mathbf{y}_{l_2}$ are obtained using Eq. (1)), with the unit vector $\hat{\mathbf{r}}$ yields unity. We refer to the original publication for the motivation behind $D_{l_1 l_2 l_3}$ [45].

B.2 Further details on the equivariant message passing

The following provides additional details on the employed equivariant message-passing architecture. Learnable weights employed in Eqs. (4), (5), and (6) (i.e., $W^{(t)}_{kk'l_2}$, $W^{(t)}_{kZ_v}$, and $W^{(t)}_{kk'l_\xi}$ with k, k' running over feature channels) are initialized by picking the respective entries from a normal distribution with zero mean and unit variance.

Many-body message-passing. The many-body equivariant features, represented by an irreducible Cartesian tensor of rank L and obtained through the ν-fold tensor product in Eq. (6), are combined using the linear expansion

$$\left(\mathbf{m}^{(t)}_{ukL}\right)_{i_1 i_2 \cdots i_L} = \Big(\sum_{\nu} \sum_{\eta_\nu} W^{(t)}_{Z_u \eta_\nu k L} \mathbf{B}^{(t)}_{u \eta_\nu k L} \Big)_{i_1 i_2 \cdots i_L}, \tag{A1}$$

where $W^{(t)}_{Z_u \eta_\nu k L}$ denotes a learnable weight matrix which depends on the chemical element Z_u and rank L and which elements are initialized by picking the respective entries from a normal distribution with zero mean and a standard deviation of $1/\mathrm{len}(\eta_\nu)$. The updated node embeddings are further obtained as a linear function of $\left(\mathbf{m}^{(t)}_{ukL}\right)_{i_1 i_2 \cdots i_L}$ and the residual connection [33, 83]

$$\left(\mathbf{h}^{(t+1)}_{ukL}\right)_{i_1 i_2 \cdots i_L} = \frac{1}{\sqrt{d_t}} \sum_{k'} W^{(t)}_{kk'L} \left(\mathbf{m}^{(t)}_{uk'L}\right)_{i_1 i_2 \cdots i_L} + \frac{1}{\sqrt{d_t N_Z}} \sum_{k'} W^{(t)}_{Z_u k k' L} \left(\mathbf{h}^{(t)}_{uk'L}\right)_{i_1 i_2 \cdots i_L}, \tag{A2}$$

where N_Z denotes the number of atom types and learnable weights $W^{(t)}_{kk'L}$ and $W^{(t)}_{Z_u k k' L}$ are initialized by picking the respective entries from a normal distribution with zero mean and unit variance.

Full and symmetric product basis. In the MACE architecture, the authors pre-compute products between the generalized Clebsch–Gordan coefficients, which define the interactions of $\{l_1, \cdots, l_\nu\}$-rank spherical tensors, and the learnable weight $W^{(t)}_{Z_u \eta_\nu k L}$ of the linear expansion in Eq. (A1) [33]. This approach reduces the computational cost of constructing the product basis with spherical tensors as the effective number of evaluated tensor products is smaller by $\mathcal{K} = \mathrm{len}\,(\eta_\nu)$. For irreducible Cartesian tensors, operations like matrix-vector and matrix-matrix products define the interactions between $\{l_1, \cdots, l_\nu\}$-rank tensors. Thus, an equivalent operation to those proposed for spherical tensors may be generally impossible for irreducible Cartesian tensors or would lead to an architecture different from MACE.

Note that one of our main goals is to demonstrate that a many-body equivariant message-passing architecture defined using higher-rank irreducible Cartesian tensors can be as expressive as the one using spherical tensors. Therefore, we compute $\mathbf{m}^{(t)}_{ukL}$ by iteratively evaluating all possible ν-fold tensor products, which effectively leads to a larger number of operations than for MACE. We refer to the models based on this architecture as irreducible Cartesian tensor potentials (ICTPs) with the full product basis, or $\mathrm{ICTP}_{\mathrm{full}}$. We also noticed that the number of tensor products $\mathrm{len}(\eta_\nu)$ leading to

the tensor of rank L, can be reduced by the symmetry of the two-fold tensor product in Eq. (6), i.e., $\mathbf{A}_{l_1} \otimes \mathbf{A}_{l_2} = \mathbf{A}_{l_2} \otimes \mathbf{A}_{l_1}$ if $\mathbf{A}_{l_1}$ and $\mathbf{A}_{l_2}$ are symmetric. We refer to this design choice as ICTPs with the symmetric product basis or ICTP_{sym}.

Coupled feature channels. Using coupled feature channels instead of uncoupled ones in Eq. (A1) can improve the performance of the final model. Additionally, the number of feature channels and, thus, the overall computational cost can be reduced when constructing the product basis. However, the number of parameters and, thus, the expressive power of the model can be preserved. Specifically, we can define a linear expansion for combining many-body equivariant features as

$$\left(\mathbf{m}^{(t)}_{ukL}\right)_{i_1 i_2 \cdots i_L} = \left(\sum_{\nu} \sum_{\eta_\nu} \sum_{k'} W^{(t)}_{Z_u \eta_\nu k k' L} \mathbf{B}^{(t)}_{u \eta_\nu k' L}\right)_{i_1 i_2 \cdots i_L}, \tag{A3}$$

where $W^{(t)}_{Z_u \eta_\nu k k' L}$ denotes a learnable weight matrix which depends on the chemical element Z_u and rank L and which elements are initialized by picking the respective entries from a normal distribution with zero mean and a standard deviation of $1/(\sqrt{d_t} \times \text{len}(\eta_\nu))$. The construction of the product basis in the latent feature space requires encoding the two-body features $\mathbf{A}^{(t)}_{ukl_\xi}$ using the learnable weight matrices in Eq. (6). The linear expansion is then constructed in the latent feature space and decoded using the learnable weight matrices of the linear function in Eq. (A2). We refer to this design choice as ICTPs with the product basis constructed in the latent feature space or ICTP_{lt}. Combining it with the symmetric product basis, we obtain $\text{ICTP}_{\text{sym+lt}}$.

Readout. Atomic energies expanded into a series of many-body contributions, $E_u = E^{(0)}_u + E^{(1)}_u + \cdots + E^{(T)}_u$, are obtained by applying readout functions $\mathcal{R}_t$ to node features with $L = 0$, which are invariant to rotations

$$E^{(t)}_u = \mathcal{R}_t\left(\left\{\mathbf{h}^{(t)}_{uk(L=0)}\right\}_k\right) = \begin{cases} \frac{1}{\sqrt{d_t}} \sum_{k'} W^{(t)}_{k'} h^{(t)}_{uk'(L=0)} & \text{if } t < T, \\ \text{NN}^{(t)}\left(\left\{\mathbf{h}^{(t)}_{uk(L=0)}\right\}_k\right) & \text{if } t = T, \end{cases} \tag{A4}$$

with learnable weights $W^{(t)}_{k'}$ or those of $\text{NN}^{(t)}$ initialized by picking the respective entries from a normal distribution with zero mean and unit variance. Linear readout functions for $t < T$ preserve the many-body orders in $\mathbf{h}^{(t)}_{uk(L=0)}$, while a one-layer fully-connected NN is used for the last message-passing layer and accounts for the residual higher-order terms in the expansion [32].

B.3 Computational complexity and runtime analysis

We analyze the computational complexity of the irreducible Cartesian tensor product with respect to the maximum rank L used. For each tuple of $(i_1, \cdots, i_L)$ of a rank-L tensor in Eq. (2), the single contraction term $(\mathbf{x}_L \cdot (k+m) \cdot \mathbf{y}_L)$ has a cost of 3^{k+m}. The total computational cost is 3^L for even tensor products and 3^{L+1} for odd ones—due to the additional double contraction with the Levi-Civita symbol. The computation of the set of permutations over the L unsymmetrized indices scales as $L!/\left(2^{L/2}\,(L/2)!\right)$ for even tensor products and $L!/\left(2^{(L-1)/2}\,((L-1)/2)!\right)$ for odd ones [45]. Because each final rank-L tensor has 3^L elements, the complexity for computing an irreducible Cartesian tensor of rank L is $\mathcal{O}\left(9^L L!/\left(2^{L/2}\,(L/2)!\right)\right)$. This expression is used throughout this work as it captures contributions from tensor contractions and index permutations, though it simplifies asymptotically to $\mathcal{O}\left(L^L\right)$ using Stirling's approximation for factorials.

The number of calculations required to obtain a rank-L spherical tensor through the Clebsch–Gordan tensor product is $(2L+1)^5$. Thus, the computational complexity of the Clebsch–Gordan tensor product is $\mathcal{O}\left(L^5\right)$. While the Clebsch–Gordan tensor product is more computationally efficient than the irreducible Cartesian tensor product for $L \to \infty$, the latter is expected to be advantageous for smaller tensor ranks, i.e., $L \leq 4$, assuming similar multiplicative factors and negligible sub-leading terms. Tensors of rank $L \leq 4$ are particularly relevant for equivariant message-passing architectures and representing physical properties [32, 33, 72]. Compared to the Gaunt tensor product with the computational complexity of $\mathcal{O}\left(L^3\right)$, including all $(l_1, l_2) \to l_3$ [36], the irreducible Cartesian tensor product may be less advantageous for $L \geq 3$. However, the Gaunt-coefficients-based approach excludes odd tensor products, limiting the expressive power and the range of possible architectures.

As regards the cost of performing message passing at each layer, the general neighbors' aggregation of Eq. (4) has a cost of $\mathcal{E} N_{\text{ch}} 9^L L!/\left(2^{L/2}\,(L/2)!\right)$, where $\mathcal{E}$ is the number of edges in the atomic

 https://doi.org/10.52202/079017-3940

system and N_{ch} is the number of feature channels. Computing many-body features via Eq. (6) requires to iteratively perform $\nu - 1$ Cartesian tensor products for all $\mathcal{K} = \text{len}(\eta_\nu)$ possible ν-fold tensor products, resulting in a computational cost of $\mathcal{M}N_{\text{ch}}\mathcal{K}(9^L L!/(2^{L/2}(L/2)!))^{\nu-1}$ with $\mathcal{M}$ denoting the number of nodes. For the Clebsch–Gordan tensor product, the corresponding number of calculations is $\mathcal{E}N_{\text{ch}}L^5$ and $\mathcal{M}N_{\text{ch}}\mathcal{K}L^{5(\nu-1)}$. Spherical models, however, can use generalized Clebsch–Gordan coefficients, resulting in $\mathcal{M}N_{\text{ch}}\mathcal{K}L^{\frac{1}{2}\nu(\nu+3)}$ for the product basis. We can remove the factor $\mathcal{K}$ from the above expression by restricting the parameterization to uncoupled feature channels as in Eq. (A1) and obtain $\mathcal{M}N_{\text{ch}}L^{\frac{1}{2}\nu(\nu+3)}$. Thus, MACE with this specific choice of the product basis can be more computationally efficient than ICTP only for large N_{ch} and small ν, given $L \leq 4$. In this work, we have shown that leveraging the symmetry of the irreducible Cartesian tensor product and coupled feature channels can improve the computational cost of ICTP. The former, in particular, reduces the effective number of ν-fold tensor products $\mathcal{K}$. Further optimization of the architecture, however, is possible and is expected to fully exploit the advantage of Cartesian tensors.

Finally, memory consumption for Cartesian tensors is often believed to be less advantageous than that of spherical tensors. However, for contracting two rank-L spherical tensors, the memory requirement is about $(2L+1)^2$, resulting from the definition of the Clebsch–Gordan tensor product in Section 2. For a Cartesian tensor, the memory requirement is about 3^L. Thus, the irreducible Cartesian tensor product can be expected to require the same or less memory for $L \leq 4$ than the Clebsch–Gordan counterpart. The memory consumption of models based on spherical tensors is further increased by the use of generalized Clebsch–Gordan coefficients, as we demonstrate in Section 5.

C Proof of Cartesian message-passing equivariance to actions of the orthogonal group

We first recap the standard action of the $\mathrm{O}(3)$ group onto $\left(\mathbb{R}^3\right)^{\otimes l}$. Let $D_{\mathcal{X}}$ be a representation of the orthogonal group $\mathrm{O}(3)$ (also in line with Section 2), i.e.,

$$D_{\mathcal{X}} : \mathrm{O}(3) \to \mathbb{R}^{3\times 3}, \quad g \mapsto D_{\mathcal{X}}[g] = R.$$

We define the action of the $\mathrm{O}(3)$ group onto $\left(\mathbb{R}^3\right)^{\otimes l}$ as follows

$$\phi : \mathrm{O}(3) \times \left(\mathbb{R}^3\right)^{\otimes l} \to \left(\mathbb{R}^3\right)^{\otimes l}, \quad \phi\left(g, \mathbf{T}_l\right)_{i_1 i_2 \cdots i_l} \coloneqq \sum_{j_1} \cdots \sum_{j_l} R_{i_1 j_1} \cdots R_{i_l j_l} \left(\mathbf{T}_l\right)_{j_1 \cdots j_l},$$

where $\mathbf{T}_l \in \left(\mathbb{R}^3\right)^{\otimes l}$. We hereinafter denote $\phi\left(g, \mathbf{T}_l\right)$ also by $R\mathbf{T}_l$. The outer product $\otimes$ is equivariant to this action, and R acts trivially on the 3×3-identity matrix.

Lemma C.1. *Let R be a representation of an element of the orthogonal group $\mathrm{O}(3)$. Then,*

(i) $\forall\, \mathbf{T}_{l_1} \in \left(\mathbb{R}^3\right)^{\otimes l_1}$ *and* $\forall\, \mathbf{T}_{l_2} \in \left(\mathbb{R}^3\right)^{\otimes l_2}$

$$R\left(\mathbf{T}_{l_1} \otimes \mathbf{T}_{l_2}\right) = \left(R\mathbf{T}_{l_1}\right) \otimes \left(R\mathbf{T}_{l_2}\right).$$

(ii) For the 3×3-identity matrix $\mathbf{I}$, we have

$$R\mathbf{I} = \mathbf{I}.$$

Proof. (i) We first show that $R\left(\mathbf{T}_{l_1} \otimes \mathbf{T}_{l_2}\right) = \left(R\mathbf{T}_{l_1}\right) \otimes \left(R\mathbf{T}_{l_2}\right)$

$$\begin{aligned}
&\left(\left(R\mathbf{T}_{l_1}\right) \otimes \left(R\mathbf{T}_{l_2}\right)\right)_{i_1 \cdots i_{l_1} i_{l_1+1} \cdots i_{l_1+l_2}} \\
&\quad = \left(R\mathbf{T}_{l_1}\right)_{i_1 \cdots i_{l_1}} \left(R\mathbf{T}_{l_2}\right)_{i_{l_1+1} \cdots i_{l_1+l_2}} \\
&\quad = \left(\sum_{j_1, \ldots, j_{l_1}} R_{i_1 j_1} \cdots R_{i_l j_{l_1}} \left(\mathbf{T}_{l_1}\right)_{j_1 \cdots j_{l_1}}\right) \left(\sum_{j_{l_1+1}, \ldots, j_{l_1+l_2}} R_{i_{l_1+1} j_{l_1+1}} \cdots R_{i_{l_1+l_2} j_{l_1+l_2}} \left(\mathbf{T}_{l_2}\right)_{j_{l_1+1} \cdots j_{l_1+l_2}}\right) \\
&\quad = \sum_{j_1} \cdots \sum_{j_{l_1}} \sum_{j_{l_1+1}} \cdots \sum_{j_{l_1+l_2}} R_{i_1 j_1} \cdots R_{i_{l_1} j_{l_1}} R_{i_{l_1+1} j_{l_1+1}} \cdots R_{i_{l_1+l_2} j_{l_1+l_2}} \left(\mathbf{T}_{l_1}\right)_{j_1 \cdots j_{l_1}} \left(\mathbf{T}_{l_2}\right)_{j_{l_1+1} \cdots j_{l_1+l_2}} \\
&\quad = \left(R\left(\mathbf{T}_{l_1} \otimes \mathbf{T}_{l_2}\right)\right)_{i_1 \cdots i_{l_1} i_{l_1+1} \cdots i_{l_1+l_2}}.
\end{aligned}$$

(ii) With $\sum_i R_{ij}R_{ik} = \delta_{jk}$, or equivalently $R^T R = \mathbf{I}$, we get

$$(R\mathbf{I})_{i_1 i_2} = \sum_{j_1, j_2} R_{i_1 j_1} R_{i_2 j_2} \delta_{j_1 j_2} = \sum_{j_1} R_{i_1 j_1} R_{i_2 j_1} = \delta_{i_1 i_2}.$$

□

Using Lemma C.1, we can show that the irreducible Cartesian tensor operator $\mathbf{T}_l : \mathbb{R}^3 \to (\mathbb{R}^3)^{\otimes l}$ is equivariant to actions of the $\mathrm{O}(3)$ group.

Proposition C.2. *Let $l \geq 0$ be a positive integer. Then, the irreducible Cartesian tensor operator $\mathbf{T}_l : \mathbb{R}^3 \to (\mathbb{R}^3)^{\otimes l}$ is equivariant to actions of the $\mathrm{O}(3)$ group.*

Proof. It suffices to show that the map $\hat{\mathbf{r}} \mapsto \hat{\mathbf{r}}^{\otimes(l-2m)} \otimes \mathbf{I}^{\otimes m}$ is equivariant to actions of the $\mathrm{O}(3)$ group $\forall\, l \geq 1$ and $\forall\, m \leq \lfloor l/2 \rfloor$ ($m \geq 0$).

$$\begin{aligned}(R\hat{\mathbf{r}})^{\otimes(l-2m)} \otimes \mathbf{I}^{\otimes m} &\overset{\text{Lemma C.1 (ii)}}{=} (R\hat{\mathbf{r}})^{\otimes(l-2m)} \otimes (R\mathbf{I})^{\otimes m} \\ &\overset{\text{Lemma C.1 (i)}}{=} R\left(\hat{\mathbf{r}}^{\otimes(l-2m)} \otimes \mathbf{I}^{\otimes m}\right).\end{aligned}$$

□

Our next claim is that for $l_3 \in \{|l_1 - l_2|, |l_1 - l_2| + 1, \cdots, l_1 + l_2\}$ the irreducible Cartesian tensor product $\otimes_{\mathrm{Cart}}$ defined in Eqs. (2) and (3) is equivariant to actions of the $\mathrm{O}(3)$ group, i.e., the following diagram commutes

$$\begin{array}{ccc} \left(\mathbb{R}^3\right)^{\otimes l_1} \times \left(\mathbb{R}^3\right)^{\otimes l_2} & \xrightarrow{\otimes_{\mathrm{Cart}}} & \left(\mathbb{R}^3\right)^{\otimes l_3} \\ \Big\downarrow R & & \Big\downarrow R \\ \left(\mathbb{R}^3\right)^{\otimes l_1} \times \left(\mathbb{R}^3\right)^{\otimes l_2} & \xrightarrow[\otimes_{\mathrm{Cart}}]{} & \left(\mathbb{R}^3\right)^{\otimes l_3} \end{array}$$

The proof for this claim is very similar to Proposition C.2. The only difference is to show the equivariance for the $(k+m)$-fold tensor contraction.

Proposition C.3. *The irreducible Cartesian tensor product $\otimes_{\mathrm{Cart}} : \left(\mathbb{R}^3\right)^{\otimes l_1} \times \left(\mathbb{R}^3\right)^{\otimes l_2} \to \left(\mathbb{R}^3\right)^{\otimes l_3}$ makes the above diagram commute.*

Proof. It suffices to show that the $(k+m)$-fold tensor contraction is equivariant to actions of the $\mathrm{O}(3)$ group. For $\mathbf{x}_{l_1} \in (\mathbb{R}^3)^{\otimes l_1}$, $\mathbf{y}_{l_2} \in (\mathbb{R}^3)^{\otimes l_2}$, and R being a representation of an element of the orthogonal group $\mathrm{O}(3)$, we can write

$$\begin{aligned}&((R\mathbf{x}_{l_1}) \cdot (s) \cdot (R\mathbf{y}_{l_2}))_{\beta_1 \cdots \beta_{l_1-s} \delta_1 \cdots \delta_{l_2-s}} \\ &\quad= \sum_{\alpha_1, \cdots, \alpha_s} (R\mathbf{x}_{l_1})_{\alpha_1 \cdots \alpha_s \beta_1 \cdots \beta_{l_1-s}} (R\mathbf{y}_{l_2})_{\alpha_1 \cdots \alpha_s \delta_1 \cdots \delta_{l_2-s}} \\ &\quad= \sum_{\alpha_1, \cdots, \alpha_s} \left(\sum_{\substack{\gamma_1, \cdots, \gamma_s \\ \eta_1, \cdots, \eta_{l_1-s}}} R_{\alpha_1 \gamma_1} \cdots R_{\alpha_s \gamma_s} R_{\beta_1 \eta_1} \cdots R_{\beta_{l_1-s} \eta_{l_1-s}} (\mathbf{x}_{l_1})_{\gamma_1 \cdots \gamma_s \eta_1 \cdots \eta_{l_1-s}} \right) \\ &\qquad\qquad \times \left(\sum_{\substack{\widetilde{\gamma}_1, \cdots, \widetilde{\gamma}_s \\ \widetilde{\eta}_1, \cdots, \widetilde{\eta}_{l_2-s}}} R_{\alpha_1 \widetilde{\gamma}_1} \cdots R_{\alpha_s \widetilde{\gamma}_s} R_{\delta_1 \widetilde{\eta}_1} \cdots R_{\delta_{l_2-s} \widetilde{\eta}_{l_2-s}} (\mathbf{y}_{l_2})_{\widetilde{\gamma}_1 \cdots \widetilde{\gamma}_s \widetilde{\eta}_1 \cdots \widetilde{\eta}_{l_2-s}} \right) \\ &\quad= \left(\sum_{\eta_1 \cdots \eta_{l_1-s}} R_{\beta_1 \eta_1} \cdots R_{\beta_{l_1-s} \eta_{l_1-s}} \right) \left(\sum_{\widetilde{\eta}_1 \cdots \widetilde{\eta}_{l_2-s}} R_{\delta_1 \widetilde{\eta}_1} \cdots R_{\delta_{l_2-s} \widetilde{\eta}_{l_2-s}} \right) \\ &\qquad\qquad \times \sum_{\gamma_1 \cdots \gamma_s} (\mathbf{x}_{l_1})_{\gamma_1 \cdots \gamma_s \eta_1 \cdots \eta_{l_1-s}} (\mathbf{y}_{l_2})_{\gamma_1 \cdots \gamma_s \widetilde{\eta}_1 \cdots \widetilde{\eta}_{l_2-s}} \\ &\quad= \left(R\left(\mathbf{x}_{l_1} \cdot (s) \cdot \mathbf{y}_{l_2}\right)\right)_{\beta_1 \cdots \beta_{l_1-s} \delta_1 \cdots \delta_{l_2-s}}.\end{aligned}$$

 https://doi.org/10.52202/079017-3940

A similar derivation applies to the odd case (3), which completes the proof. □

We finally give a proof for Proposition 4.1, i.e., the equivariance of message-passing layers based on irreducible Cartesian tensors to actions of the $\mathrm{O}(3)$ group.

Proof of Proposition 4.1. Since a message-passing layer in Section 4.2 and Appendix B.2 is defined by stacking layers corresponding to Eqs. (4), (5), (6), (A1), (A2), and (A3), we show the equivariance for each of these equations. We show the equivariance by induction.

- Case $t = 1$:

  Equivariance of Eq. (5): Since learnable weights W_{kZ_v} and learnable (invariant) radial basis $R^{(1)}_{kl_1}(r_{uv})$ are defined for each feature channel k, the weights are multiplied with $\big(\mathbf{T}_{l_1}(\hat{\mathbf{r}}_{uv})\big)_{i_1 i_2 \cdots i_{l_1}}$ as scalars. Therefore,

  $$\big(\mathbf{A}^{(1)}_{ukl_1}\big)_{i_1 i_2 \cdots i_{l_1}} = \sum\nolimits_{v \in \mathcal{N}(u)} R^{(1)}_{kl_1}(r_{uv}) \big(\mathbf{T}_{l_1}(\hat{\mathbf{r}}_{uv})\big)_{i_1 i_2 \cdots i_{l_1}} W_{kZ_v}$$

  is equivariant to actions of the $\mathrm{O}(3)$ group.
  Equivariance of Eq. (6): The proof is similar to the case of Eq. (5), since we again have learnable parameter $W^{(1)}_{kk'l_\nu}$ for each $\mathbf{A}^{(1)}_{uk'l_\nu}$. Therefore, noting that the Cartesian tensor product is equivariant to actions of the $\mathrm{O}(3)$ group, the following function

  $$\big(\mathbf{B}^{(1)}_{u\eta_\nu kL}\big)_{i_1 i_2 \cdots i_L} = \big(\tilde{\mathbf{A}}^{(1)}_{ukl_1} \otimes_{\mathrm{Cart}} \cdots \otimes_{\mathrm{Cart}} \tilde{\mathbf{A}}^{(1)}_{ukl_\nu}\big)_{i_1 i_2 \cdots i_L}$$

  is also equivariant to actions of the $\mathrm{O}(3)$ group.
  Equivariance of Eq. (A1): The equation defined by

  $$\big(\mathbf{m}^{(1)}_{ukL}\big)_{i_1 i_2 \cdots i_L} = \Big(\sum_{\nu} \sum_{\eta_\nu} W^{(1)}_{Z_u \eta_\nu kL} \mathbf{B}^{(1)}_{u\eta_\nu kL}\Big)_{i_1 i_2 \cdots i_L},$$

  is equivariant to actions of the $\mathrm{O}(3)$ group, since for each set of indices u, η_ν, k, and L we have a scalar learnable parameter $W^{(1)}_{Z_u \eta_\nu kL}$.
  Equivariance of Eq. (A2): The respective parameters $W^{(1)}_{kk'L}$ and $W^{(1)}_{Z_u kk'L}$ in Eq. (A2)

  $$\big(\mathbf{h}^{(2)}_{ukL}\big)_{i_1 i_2 \cdots i_L} = \frac{1}{\sqrt{d_t}} \sum_{k'} W^{(1)}_{kk'L} \big(\mathbf{m}^{(1)}_{uk'L}\big)_{i_1 i_2 \cdots i_L} + \frac{1}{\sqrt{d_t N_Z}} \sum_{k'} W^{(1)}_{Z_u kk'L} \big(\mathbf{h}^{(1)}_{uk'L}\big)_{i_1 i_2 \cdots i_L}.$$

  are multiplied as scalar with $\mathbf{h}^{(1)}_{uk'L}$ and $\mathbf{m}^{(1)}_{uk'L}$, which are shown to be equivariant to actions of the $\mathrm{O}(3)$ group. Thus, $\mathbf{h}^{(2)}_{ukL}$ is equivariant to actions of the $\mathrm{O}(3)$ group as a function of $\mathbf{h}^{(1)}_{uk'L}$ and $\mathbf{m}^{(1)}_{uk'L}$.
  Equivariance of Eq. (A3): The equivariance of Eq. (A3) to actions of the $\mathrm{O}(3)$ group is also immediate since learnable parameters apply to each tuple of $(i_1, i_2, \cdots, i_L)$ as a scalar.

- General case $t > 1$: The equivariance of Eqs. (4), (5), (6), (A1), (A2), and (A3) to actions of the $\mathrm{O}(3)$ group for arbitrary $t > 1$ follows in a similar way to those obtained for the $t = 1$ case. However, we need to show the equivariance of Eq. (4). Suppose that Eqs. (5), (6), (A1), (A2), and (A3) are equivariant to actions of the $\mathrm{O}(3)$ group for $t > 1$. Then, for the $(t+1)$-th message-passing layer,

  $$\big(\mathbf{A}^{(t)}_{ukl_3}\big)_{i_1 i_2 \cdots i_{l_3}} = \sum\nolimits_{v \in \mathcal{N}(u)} \Big(R^{(t)}_{kl_1 l_2 l_3}(r_{uv}) \mathbf{T}_{l_1}(\hat{\mathbf{r}}_{uv}) \otimes_{\mathrm{Cart}} \frac{1}{\sqrt{d_t}} \sum_{k'} W^{(t)}_{kk'l_2} \mathbf{h}^{(t)}_{vk'l_2}\Big)_{i_1 i_2 \cdots i_{l_3}},$$

  is equivariant to actions of the $\mathrm{O}(3)$ group. Here, the irreducible Cartesian tensor product $\otimes_{\mathrm{Cart}}$ and $\mathbf{T}_{l_1}$ are equivariant to actions of the $\mathrm{O}(3)$ group, $\mathbf{h}^{(t)}_{vk'l_2}$ is equivariant by the assumption, and $W^{(t)}_{kk'l_2}$ applies to $\mathbf{h}^{(t)}_{vk'l_2}$ as a scalar.

□

D Proof of the traceless property for irreducible Cartesian tensors

This section provides a proof for Proposition 4.2 in the main text. The corresponding proposition states the following

Proposition D.1. *The message-passing layers based on irreducible Cartesian tensors and their irreducible tensor products preserve the traceless property of irreducible Cartesian tensors.*

Our proof of Proposition 4.2 comprises two main parts. First, we show that given two irreducible Cartesian tensors, the irreducible Cartesian tensor product yields again an irreducible Cartesian tensor. We then can use this result in the second part, where we show that expressions defined by Eqs. (4), (5), (6), (A1), (A2), and (A3) preserve the traceless property of irreducible Cartesian tensors. The proof of the second part is straightforward since the message-passing layers are defined by multiplications with scalars, summations, and the irreducible Cartesian tensor product of tensors defined in Eqs. (1), (2), and (3). Therefore, we provide the proof only for the first part in the rest of this section.

Irreducible Cartesian tensors from unit vectors. Recall that an irreducible Cartesian tensor of arbitrary rank l for a unit vector $\hat{\mathbf{r}} \in \mathbb{R}^3$ is defined as

$$\mathbf{T}_l\left(\hat{\mathbf{r}}\right) = C \sum\nolimits_{m=0}^{\lfloor l/2 \rfloor} (-1)^m \frac{(2l-2m-1)!!}{(2l-1)!!} \left\{\hat{\mathbf{r}}^{\otimes(l-2m)} \otimes \mathbf{I}^{\otimes m}\right\}.$$

The trace of this tensor reads

$$\begin{aligned}
\mathrm{Tr}\left(\mathbf{T}_l\left(\hat{\mathbf{r}}\right)\right) &= \sum\nolimits_{i_1=i_2} \left(\mathbf{T}_l\left(\hat{\mathbf{r}}\right)\right)_{i_1 i_2} \\
&= \sum\nolimits_{i_1=i_2} C \sum\nolimits_{m=0}^{\lfloor l/2 \rfloor} (-1)^m \frac{(2l-2m-1)!!}{(2l-1)!!} \left(\left\{\hat{\mathbf{r}}^{\otimes(l-2m)} \otimes \mathbf{I}^{\otimes m}\right\}\right)_{i_1 i_2} \\
&= C \sum\nolimits_{m=0}^{\lfloor l/2 \rfloor} (-1)^m \frac{(2l-2m-1)!!}{(2l-1)!!} \sum\nolimits_{i_1=i_2} \left(\left\{\hat{\mathbf{r}}^{\otimes(l-2m)} \otimes \mathbf{I}^{\otimes m}\right\}\right)_{i_1 i_2}.
\end{aligned}$$

Let $\left\{k_1, k_2, \ldots, k_{l-2m}\right\}$ be a subset of $\{1, 2, \ldots, l\}$ with $l-2m$ distinct elements, and a subset I_{l-2m} with $l-2m$ distinct elements in $\left\{i_1, \ldots, i_l\right\}$ is written as $I_{l-2m} = \left\{i_{k_1}, i_{k_2}, \ldots, i_{k_{l-2m}}\right\}$. We also let J_{l-2m} be a set of m disjoint subsets with 2 elements in $\left\{i_1, \ldots, i_l\right\} \backslash I_{l-2m}$, i.e., $J_{l-2m} = \{\{i_{k_{l-2m+1}}, i_{k_{l-2m+2}}\}, \ldots, \{i_{k_{l-1}}, i_{k_l}\}\}$. While we introduce I_{l-2m} and J_{l-2m} as sets, we assume the order of elements in those sets if the order is necessary and there is no confusion. Furthermore, we introduce two notations used throughout this section

$$\begin{aligned}
\hat{\mathbf{r}}_{I_{l-2m}}^{\otimes(l-2m)} &= \hat{r}_{i_{k_1}} \hat{r}_{i_{k_2}} \cdots \hat{r}_{i_{k_{l-2m}}} \\
\mathbf{I}_{J_{l-2m}}^{\otimes m} &= \delta_{i_{k_{l-2m+1}} i_{k_{l-2m+2}}} \cdots \delta_{i_{k_{l-1}} i_{k_l}}.
\end{aligned}$$

Then, we can write

$$\begin{aligned}
\left\{\hat{\mathbf{r}}^{\otimes(l-2m)} \otimes \mathbf{I}^{\otimes m}\right\} &= \sum\nolimits_{I_{l-2m}} \sum\nolimits_{J_{l-2m}} \hat{\mathbf{r}}_{I_{l-2m}}^{\otimes(l-2m)} \otimes \mathbf{I}_{J_{l-2m}}^{\otimes m} \\
&= \sum\nolimits_{I_{l-2m}} \sum\nolimits_{J_{l-2m}} \hat{r}_{i_{k_1}} \cdots \hat{r}_{i_{k_{l-2m}}} \delta_{i_{k_{l-2m+1}} i_{k_{l-2m+2}}} \cdots \delta_{i_{k_{l-1}} i_{k_l}},
\end{aligned}$$

where I_{l-2m} runs over all subsets of $\left\{i_1, \ldots, i_l\right\}$ with $l-2m$ elements and J_{l-2m} comprises all $\left(\prod_{p=m}^{1} \binom{2p}{2}\right)/m!$ combinations of m subsets with 2 elements of $\left\{i_1, \ldots, i_l\right\} \backslash I_{l-2m}$. Depending on whether i_1 and/or i_2 belong to I_{l-2m}, the above expression may be further reduced to

$$\begin{aligned}
\Bigg(&\overset{(a)_m}{\sum_{i_1, i_2 \in I_{l-2m}}} + \overset{(b)_m}{\sum_{\substack{i_1 \in I_{l-2m} \\ i_2 \in J_{l-2m}}}} + \overset{(c)_m}{\sum_{\substack{i_2 \in I_{l-2m} \\ i_1 \in J_{l-2m}}}} + \overset{(d)_m}{\sum_{\substack{i_1, i_2 \notin I_{l-2m} \\ \delta_{i_1 i_2}}}} + \overset{(e)_m}{\sum_{\substack{i_1, i_2 \notin I_{l-2m} \\ \delta_{i_s i_1} \delta_{i_2 i_t}}}} \Bigg) \hat{r}_{i_{k_1}} \cdots \hat{r}_{i_{k_{l-2m}}} \\
&\times \delta_{i_{k_{l-2m+1}} i_{k_{l-2m+2}}} \cdots \delta_{i_{k_{l-1}} i_{k_l}}.
\end{aligned}$$

 https://doi.org/10.52202/079017-3940

We get the following traces for each pair of I_{l-2m} and J_{l-2m}

$$\begin{aligned}
\mathrm{Tr}\left((a)_m\right) &= \hat{\mathbf{r}}^{\otimes(l-2m)}_{I_{l-2m}\setminus\{i_1,i_2\}} \otimes \mathbf{I}^{\otimes m}_{J_{l-2m}}, \\
\mathrm{Tr}\left((b)_m\right) &= \sum\nolimits_{i_1=i_2} \hat{r}_{i_{k_1}} \cdots \hat{r}_{i_1} \cdots \hat{r}_{i_{k_{l-2m}}} \delta_{i_{k_{l-2m+1}} i_{k_{l-2m+2}}} \cdots \delta_{i_{k_{l-1}} i_{k_l}} \delta_{i_{k_s} i_2} \\
&= \hat{r}_{i_{k_1}} \cdots \hat{r}_{i_{k_s}} \cdots \hat{r}_{i_{k_{l-2m}}} \delta_{i_{k_{l-2m+1}} i_{k_{l-2m+2}}} \cdots \delta_{i_{k_{l-1}} i_{k_l}} \\
&= \hat{\mathbf{r}}^{\otimes(l-2m)}_{\{i_{k_s}\}\cup I_{l-2m}\setminus\{i_1\}} \otimes \mathbf{I}^{\otimes(m-1)}_{J_{l-2m}\setminus\{i_{k_s},i_2\}}, \\
\mathrm{Tr}\left((c)_m\right) &= \sum\nolimits_{i_1=i_2} \hat{r}_{i_{k_1}} \cdots \hat{r}_{i_2} \cdots \hat{r}_{i_{k_{l-2m}}} \delta_{i_{k_{l-2m+1}} i_{k_{l-2m+2}}} \cdots \delta_{i_{k_{l-1}} i_{k_l}} \delta_{i_{k_s} i_1} \\
&= \hat{r}_{i_{k_1}} \cdots \hat{r}_{i_{k_s}} \cdots \hat{r}_{i_{k_{l-2m}}} \delta_{i_{k_{l-2m+1}} i_{k_{l-2m+2}}} \cdots \delta_{i_{k_{l-1}} i_{k_l}} \\
&= \hat{\mathbf{r}}^{\otimes(l-2m)}_{\{i_{k_s}\}\cup I_{l-2m}\setminus\{i_2\}} \otimes \mathbf{I}^{\otimes(m-1)}_{J_{l-2m}\setminus\{i_{k_s},i_1\}}, \\
\mathrm{Tr}\left((d)_m\right) &= \sum\nolimits_{i_1=i_2} \hat{r}_{i_{k_1}} \cdots \hat{r}_{i_{k_{l-2m}}} \delta_{i_{k_{l-2m+1}} i_{k_{l-2m+2}}} \cdots \delta_{i_{k_{l-1}} i_{k_l}} \delta_{i_1 i_2} \\
&= 3\left(\hat{\mathbf{r}}^{\otimes(l-2m)}_{I_{l-2m}} \otimes \mathbf{I}^{\otimes(m-1)}_{J_{l-2m}\setminus\{i_1,i_2\}}\right), \\
\mathrm{Tr}\left((e)_m\right) &= \sum\nolimits_{i_1=i_2} \hat{r}_{i_{k_1}} \ldots \hat{r}_{i_{k_{l-2m}}} \delta_{i_{k_{l-2m+1}} i_{k_{l-2m+2}}} \cdots \delta_{i_s i_1} \delta_{i_2 i_t} \cdots \delta_{i_{k_{l-1}} i_{k_l}} \\
&= \hat{\mathbf{r}}^{\otimes(l-2m)}_{I_{l-2m}} \otimes \mathbf{I}^{\otimes(m-1)}_{J_{l-2m}\setminus\{i_1,i_2\}}.
\end{aligned}$$

Lemma D.2. *Let $l \geq 2$ and $0 \leq m \leq \lfloor l/2 \rfloor - 1$. For each pair (I_{l-2m}, J_{l-2m}), there exist $2l-2m-1$ pairs of $\left(I_{l-2(m+1)}, J_{l-2(m+1)}\right)$ whose trace equals to the trace for the pair (I_{l-2m}, J_{l-2m}), i.e.,*

$$(2l-2m-1)\,\mathrm{Tr}\left(\hat{\mathbf{r}}^{\otimes(l-2m)}_{I_{l-2m}} \otimes \mathbf{I}^{\otimes m}_{J_{l-2m}}\right) = \mathrm{Tr}\left(\sum_{(I_{l-2(m+1)}, J_{l-2(m+1)})} \hat{\mathbf{r}}^{\otimes(l-2(m+1))}_{I_{l-2(m+1)}} \otimes \mathbf{I}^{\otimes(m+1)}_{J_{l-2(m+1)}}\right).$$

Proof. Without loss of generality, we may assume $I_{l-2m} = \left(i_1, i_2, i_{k_3}, \ldots, i_{k_{l-2m}}\right)$ and $J_{l-2m} = \left\{\left\{i_{k_{l-2m+1}}, i_{k_{l-2m+2}}\right\}, \ldots, \left\{i_{k_{l-1}}, i_{k_l}\right\}\right\}$. For $\forall\, m$, $\mathrm{Tr}\left((a)_m\right)$ has the following expression

$$\begin{aligned}
\mathrm{Tr}\left((a)_m\right) &= \sum\nolimits_{i_1=i_2} \hat{r}_{i_1} \hat{r}_{i_2} \hat{r}_{i_{k_3}} \cdots \hat{r}_{i_{k_{l-2m}}} \delta_{i_{k_{l-2m+1}} i_{k_{l-2m+2}}} \cdots \delta_{i_{k_{l-1}} i_{k_l}} \\
&= \hat{r}_{i_{k_3}} \cdots \hat{r}_{i_{k_{l-2m}}} \delta_{i_{k_{l-2m+1}} i_{k_{l-2m+2}}} \cdots \delta_{i_{k_{l-1}} i_{k_l}}.
\end{aligned}$$

For $\mathrm{Tr}\left((b)_{m+1}\right)$ and $m+1$, we have a set of $\left(I_{l-2(m+1)}, J_{l-2(m+1)}\right)$ with the cardinality of $l-2m-2$ and the corresponding trace equals to $\mathrm{Tr}\left((a)_m\right)$. For $3 \leq \forall\, s \leq l-2m$, let

$$\begin{aligned}
I^{(s)}_{l-2(m+1)} &= \Big\{i_{k_3}, \ldots, \underbrace{i_1}_{s\text{-th index}}, \ldots, i_{k_{l-2m}}\Big\}, \\
J^{(s)}_{l-2(m+1)} &= \left\{\left\{i_{k_{l-2m+1}}, i_{k_{l-2m+2}}\right\}, \ldots, \left\{i_{k_{l-1}}, i_{k_l}\right\}, \left\{i_{k_s}, i_2\right\}\right\}.
\end{aligned}$$

Here, we note that we assume the order of elements in $I^{(s)}_{l-2(m+1)}$. Then, we obtain

$$\begin{aligned}
\mathrm{Tr}\left((b)_{m+1}\right) &= \sum\nolimits_{i_1=i_2} \sum\nolimits_{s=3}^{l-2m} \hat{r}_{i_{k_3}} \cdots \underbrace{\hat{r}_{i_1}}_{s\text{-th vector}} \cdots \hat{r}_{i_{k_{l-2m}}} \delta_{i_{k_{l-2m+1}} i_{k_{l-2m+2}}} \cdots \delta_{i_{k_{l-1}} i_{k_l}} \delta_{i_{k_s} i_2} \\
&= \sum\nolimits_{s=3}^{l-2m} \hat{r}_{i_{k_3}} \cdots \hat{r}_{i_{k_s}} \cdots \hat{r}_{i_{k_{l-2m}}} \delta_{i_{k_{l-2m+1}} i_{k_{l-2m+2}}} \cdots \delta_{i_{k_{l-1}} i_{k_l}} \\
&= (l-2m-2)\,\mathrm{Tr}\left((a)_m\right).
\end{aligned}$$

The same derivation as above holds for $\mathrm{Tr}\left((c)_{m+1}\right)$. For $\mathrm{Tr}\left((d)_{m+1}\right)$, let

$$\begin{aligned}
I_{l-2(m+1)} &= \left\{i_{k_3}, \ldots, i_{k_{l-2m}}\right\}, \\
J_{l-2(m+1)} &= \left\{\left\{i_{k_{l-2m+1}}, i_{k_{l-2m+2}}\right\}, \ldots, \left\{i_{k_{l-1}}, i_{k_l}\right\}, \left\{i_1, i_2\right\}\right\}.
\end{aligned}$$

and we get

$$\begin{aligned}\mathrm{Tr}\left((d)_{m+1}\right) &= \sum\nolimits_{i_1=i_2} \hat{r}_{i_{k_3}} \cdots \hat{r}_{i_{k_{l-2m}}} \delta_{i_{k_{l-2m+1}} i_{k_{l-2m+2}}} \cdots \delta_{i_{k_{l-1}} i_{k_l}} \delta_{i_1 i_2} \\ &= 3\,\mathrm{Tr}\left((a)_m\right).\end{aligned}$$

For $\mathrm{Tr}\left((e)_{m+1}\right)$ and $1 \leq \forall\, s \leq m$, we let

$$\begin{aligned}I^{(s)}_{l-2(m+1)} &= \left\{i_{k_3}, \ldots, i_{k_{l-2m}}\right\}, \\ J^{(s)}_{l-2(m+1)} &= \left\{\left\{i_{k_{l-2m+1}}, i_{k_{l-2m+2}}\right\}, \ldots, \left\{i_{k_{l-2m+2s-1}}, i_1\right\}, \left\{i_2, i_{k_{l-2m+2s}}\right\}, \ldots, \left\{i_{k_{l-1}}, i_{k_l}\right\}\right\},\end{aligned}$$

and obtain

$$\begin{aligned}\mathrm{Tr}\left((e)_{m+1}\right) &= \sum\nolimits_{i_1=i_2} \sum\nolimits_{s=1}^{m} \hat{r}_{i_{k_3}} \cdots \hat{r}_{i_{k_{l-2m}}} \delta_{i_{k_{l-2m+1}} i_{k_{l-2m+2}}} \cdots \delta_{i_{k_{l-2m+2s-1}} i_1} \delta_{i_2 i_{k_{l-2m+2s}}} \cdots \delta_{i_{k_{l-1}} i_{k_l}} \\ &= \sum\nolimits_{s=1}^{m} \hat{r}_{i_{k_3}} \cdots \hat{r}_{i_{k_{l-2m}}} \delta_{i_{k_{l-2m+1}} i_{k_{l-2m+2}}} \cdots \delta_{k_{l-2m+2s-1} k_{l-2m+2s}} \cdots \delta_{i_{k_{l-1}} i_{k_l}} \\ &= 2m\,\mathrm{Tr}\left((a)_m\right).\end{aligned}$$

The additional factor of two originates from the permutation of indices i_1 and i_2.

Using all the above results, we obtain

$$\begin{aligned}&\mathrm{Tr}((b)_{m+1}) + \mathrm{Tr}((c)_{m+1}) + \mathrm{Tr}((d)_{m+1}) + \mathrm{Tr}((e)_{m+1}) \\ &\quad = (l-2m-2)\,\mathrm{Tr}\left((a)_m\right) + (l-2m-2)\,\mathrm{Tr}\left((a)_m\right) + 3\,\mathrm{Tr}\left((a)_m\right) + 2m\,\mathrm{Tr}\left((a)_m\right) \\ &\quad = (2l-2m-1)\,\mathrm{Tr}\left((a)_m\right),\end{aligned}$$

which completes the proof of the Lemma. □

For each m, define $A(m) = \mathrm{Tr}\left((a)_m\right)$ and $B(m) = \mathrm{Tr}\left((b)_m\right) + \mathrm{Tr}\left((c)_m\right) + \mathrm{Tr}\left((d)_m\right) + \mathrm{Tr}\left((e)_m\right)$. Then, the trace of $\mathbf{T}_l\left(\hat{\mathbf{r}}\right)$ may be written as a summation of $A(m)$ and $B(m)$ over m

$$\begin{aligned}\mathrm{Tr}\left(\mathbf{T}_l\left(\hat{\mathbf{r}}\right)\right) &= C \sum\nolimits_{m=0}^{\lfloor l/2 \rfloor} (-1)^m \frac{(2l-2m-1)!!}{(2l-1)!!} \sum\nolimits_{i_1=i_2} \left(\left\{\hat{\mathbf{r}}^{\otimes(l-2m)} \otimes \mathbf{I}^{\otimes m}\right\}\right)_{i_1,i_2} \\ &= C(\qquad\qquad (A(0) + B(0)) \\ &\qquad + (-1)\frac{(2l-2-1)!!}{(2l-1)!!}(A(1) + B(1)) \\ &\qquad\qquad \cdots \\ &\qquad + (-1)^{\tilde{m}} \frac{(2l-2\tilde{m}-1)!!}{(2l-1)!!}(A(\tilde{m}) + B(\tilde{m})) \\ &\quad + (-1)^{\tilde{m}+1} \frac{(2l-2(\tilde{m}+1)-1)!!}{(2l-1)!!}(A(\tilde{m}+1) + B(\tilde{m}+1)) \\ &\qquad\qquad \cdots .\end{aligned}$$

By Lemma D.2, each pair of the diagonal terms $(A(\tilde{m}), B(\tilde{m}+1))$ is cancelled, and $B(0)$ and $A(\lfloor l/2 \rfloor)$ are left, which by definition are trivial. This result completes the proof of the traceless property for Eq. (1).

Even irreducible Cartesian tensor product. The trace of an even irreducible Cartesian tensor product may be written as

$$\begin{aligned}&\mathrm{Tr}\left((\mathbf{x}_{l_1} \otimes_{\mathrm{Cart}} \mathbf{y}_{l_2})_{l_3}\right) \\ &\quad = \sum_{i_1=i_2} C_{l_1 l_2 l_3} \sum\nolimits_{m=0}^{\min(l_1,l_2)-k} (-1)^m 2^m \frac{(2l_3-2m-1)!!}{(2l_3-1)!!} \left(\left\{(\mathbf{x}_{l_1} \cdot (k+m) \cdot \mathbf{y}_{l_2}) \otimes \mathbf{I}^{\otimes m}\right\}\right)_{i_1 i_2} \\ &\quad = C_{l_1 l_2 l_3} \sum\nolimits_{m=0}^{\min(l_1,l_2)-k} (-1)^m 2^m \frac{(2l_3-2m-1)!!}{(2l_3-1)!!} \underbrace{\sum_{i_1=i_2} \left(\left\{(\mathbf{x}_{l_1} \cdot (k+m) \cdot \mathbf{y}_{l_2}) \otimes \mathbf{I}^{\otimes m}\right\}\right)_{i_1 i_2}}_{(*)_{\text{even}}}.\end{aligned}$$

https://doi.org/10.52202/079017-3940

In the following, we let

$$
\begin{aligned}
I_{l_1-(k+m)} &= \left\{i_{k_1}, \ldots, i_{k_{l_1-(k+m)}}\right\} \subset \left\{i_1, \ldots, i_{l_3}\right\}, \\
J_{l_2-(k+m)} &= \left\{i_{k_{l_1-(k+m)+1}}, \ldots, i_{k_{l_3-2m}}\right\} \subset \left\{i_1, \ldots, i_{l_3}\right\} \backslash I_{l_1-(k+m)}, \\
K_{l_3-2m} &= \left\{\left\{i_{k_{l_3-2m+1}}, i_{k_{l_3-2m+2}}\right\}, \ldots, \left\{i_{k_{l_3-1}}, i_{k_{l_3}}\right\}\right\}.
\end{aligned}
$$

For simplicity, we omit the subscripts $l_1-(k+m)$, $l_2-(k+m)$, and l_3-2m for I, J, and K, if there is no confusion. We also introduce a new notation $\mathbf{x}_{j_1\cdots j_p,Q} = \mathbf{x}_{j_1\cdots j_p,q_1\cdots q_u}$, where $\mathbf{x}$ is an irreducible Cartesian tensor and $Q = \left\{q_1, \ldots, q_u\right\}$, which is used throughout this section. Then, non-trivial terms in $(*)_{\text{even}}$ may be written as

$$
\sum_{i_1=i_2} \left(\sum_{\substack{i_1\in I \\ i_2\in J}}^{(a)_m} + \sum_{\substack{i_2\in I \\ i_1\in J}}^{(b)_m} + \sum_{\substack{i_1\in I\cup J \\ i_2\in K}}^{(c)_m} + \sum_{\substack{i_2\in I\cup J \\ i_1\in K}}^{(d)_m} + \sum_{i_1,i_2\in K}^{(e)_m} \right) \sum_{j_1,\ldots,j_{k+m}} \left(\mathbf{x}_{l_1}\right)_{j_1\cdots j_{k+m},I} \left(\mathbf{y}_{l_2}\right)_{j_1\cdots j_{k+m},J} \mathbf{I}_K^{\otimes m}.
$$

We note that the trace is trivial when both of i_1 and i_2 belong to either of I or J, because $\mathbf{x}_{l_1}$ and $\mathbf{y}_{l_2}$ are traceless by assumption.

Lemma D.3. *Let $0 \le m \le \min(l_1, l_2) - k - 1$. For each triplet $\left(I_{l_1-(k+m)}, J_{l_2-(k+m)}, K_{l_3-2m}\right)$, there exist $2l_3 - 2m - 1$ triplets of $\left(I_{l_1-(k+m+1)}, J_{l_2-(k+m+1)}, K_{l_3-2(m+1)}\right)$ such that the trace for each triplet equals to the trace for $\left(I_{l_1-(k+m)}, J_{l_2-(k+m)}, K_{l_3-2m}\right)$.*

Proof. Without loss of generality, we may assume

$$
\begin{aligned}
I_{l_1-(k+m)} &= \left\{i_1, i_{k_2}, \ldots, i_{k_{l_1-(k+m)}}\right\} \subset \left\{i_1, \ldots, i_{l_3}\right\}, \\
J_{l_2-(k+m)} &= \left\{i_2, i_{k_{l_1-(k+m)+2}}, \ldots, i_{k_{l_3-2m}}\right\} \subset \left\{i_1, \ldots, i_{l_3}\right\} \backslash I_{l_1-(k+m)}, \\
K_{l_3-2m} &= \left\{\left\{i_{k_{l_3-2m+1}}, i_{k_{l_3-2m+2}}\right\}, \ldots, \left\{i_{k_{l_3-1}}, i_{k_{l_3}}\right\}\right\}.
\end{aligned}
$$

Then, $\operatorname{Tr}\left((a)_m\right)$ has the following expression

$$
\begin{aligned}
&\operatorname{Tr}\left((a)_m\right) \\
&\quad = \sum_{i_1=i_2} \sum_{j_1,\ldots,j_{k+m}} \left(\mathbf{x}_{l_1}\right)_{j_1\cdots j_{k+m},i_1 i_{k_2}\cdots i_{k_{l_1-(k+m)}}} \left(\mathbf{y}_{l_2}\right)_{j_1\cdots j_{k+m},i_2 i_{k_{l_1-(k+m)+2}}\cdots i_{k_{l_3-2m}}} \mathbf{I}_K^{\otimes m} \\
&\quad = \sum_{\substack{j_1,\ldots,j_{k+m}, \\ j_{k+m+1}}} \left(\mathbf{x}_{l_1}\right)_{j_1\cdots j_{k+m+1},i_{k_2}\cdots i_{k_{l_1-(k+m)}}} \left(\mathbf{y}_{l_2}\right)_{j_1\cdots j_{k+m+1},i_{k_{l_1-(k+m)+2}}\cdots i_{k_{l_3-2m}}} \mathbf{I}_K^{\otimes m}.
\end{aligned}
$$

The same derivation applies to $\operatorname{Tr}\left((b)_m\right)$, and it has the same result as above.

For $\operatorname{Tr}((c)_{m+1})$ and $m+1$, we have $l_3 - 2m - 2$ of $\left(I_{l_1-(k+m+1)}, J_{l_2-(k+m+1)}, K_{l_3-2(k+m+1)}\right)$ triplets whose trace equals to $\operatorname{Tr}\left((a)_m\right)$. Indeed, like in the proof of Lemma D.2, we let

$$
\begin{aligned}
I_{l_1-(k+m+1)} \cup J_{l_2-(k+m+1)} &= \left\{i_{k_2}, \ldots, i_1, \ldots, i_{k_{l_1-(k+m)}}, i_{k_{l_1-(k+m)+2}}, \ldots, i_{k_{l_3-2m}}\right\}, \\
K_{l_3-2(m+1)} &= \left\{\left\{i_{k_{l_3-2m+1}}, i_{k_{l_3-2m+2}}\right\}, \ldots, \left\{i_{k_{l_3-1}}, i_{k_{l_3}}\right\}, \left\{i_{k_s}, i_2\right\}\right\}.
\end{aligned}
$$

Then, by letting $\tilde{J} = \left\{j_1, \ldots, j_{k+m}, j_{k+m+1}\right\}$, we obtain

$$
\begin{aligned}
&\operatorname{Tr}((c)_{m+1}) \\
&\quad = \sum_{i_1=i_2} \sum_{s} \sum_{\tilde{J}} \left(\mathbf{x}_{l_1}\right)_{\tilde{J} i_{k_2}\cdots i_1\cdots i_{k_{l_1-(k+m)}}} \left(\mathbf{y}_{l_2}\right)_{\tilde{J} i_{k_{l_1-(k+m)+2}}\cdots i_{k_{l_3-2m}}} \\
&\qquad\qquad \times \delta_{i_{k_{l_3-2m+1}} i_{k_{l_3-2m+2}}} \cdots \delta_{i_{k_{l_3-1}} i_{k_{l_3}}} \delta_{i_{k_s} i_2} \\
&\quad = \sum_{s} \sum_{\tilde{J}} \left(\mathbf{x}_{l_1}\right)_{\tilde{J},i_{k_2}\cdots i_{k_s}\cdots i_{k_{l_1-(k+m)}}} \left(\mathbf{y}_{l_2}\right)_{\tilde{J},i_{k_{l_1-(k+m)+2}}\cdots i_{k_{l_3-2m}}} \\
&\qquad\qquad \times \delta_{i_{k_{l_3-2m+1}},i_{k_{l_3-2m+2}}} \cdots \delta_{i_{k_{l_3-1}},i_{k_{l_3}}} \\
&\quad = (l_3 - 2m - 2) \operatorname{Tr}\left((a)_m\right).
\end{aligned}
$$

The same derivation applies to $\mathrm{Tr}((d)_{m+1})$, and we get $\mathrm{Tr}((d)_{m+1}) = (l_3 - 2m - 2)\,\mathrm{Tr}\left((a)_m\right)$.

For $\mathrm{Tr}((e)_{m+1})$, the same derivation as for $\mathrm{Tr}((d)_{m+1})$ and $\mathrm{Tr}((e)_{m+1})$ in the proof of Lemma D.2 applies. Therefore, we have

$$\mathrm{Tr}((e)_{m+1}) = (2m+3)\,\mathrm{Tr}\left((a)_m\right).$$

Hence, we obtain

$$\begin{aligned}
&\mathrm{Tr}((c)_{m+1}) + \mathrm{Tr}((d)_{m+1}) + \mathrm{Tr}((e)_{m+1}) \\
&\quad = (l_3 - 2m - 2)\,\mathrm{Tr}((a)_m) + (l_3 - 2m - 2)\,\mathrm{Tr}((a)_m) + (2m+3)\,\mathrm{Tr}((a)_m) \\
&\quad = (2l_3 - 2m - 1)\,\mathrm{Tr}((a)_m) = (2l_3 - 2m - 1)\,\mathrm{Tr}((b)_m),
\end{aligned}$$

which completes the proof of the Lemma. □

For each m, define $A(m) = \mathrm{Tr}\left((a)_m\right) + \mathrm{Tr}\left((b)_m\right)$ and $B(m) = \mathrm{Tr}\left((c)_m\right) + \mathrm{Tr}\left((d)_m\right) + \mathrm{Tr}\left((e)_m\right)$. Then, the trace for $\left(\mathbf{x}_{l_1} \otimes_{\mathrm{Cart}} \mathbf{y}_{l_2}\right)_{l_3}$ may be written as a summation of $A(m)$ and $B(m)$ over m

$$\begin{aligned}
&\mathrm{Tr}\left(\left(\mathbf{x}_{l_1} \otimes_{\mathrm{Cart}} \mathbf{y}_{l_2}\right)_{l_3}\right) \\
&\quad = C_{l_1 l_2 l_3} \sum\nolimits_{m=0}^{\min(l_1,l_2)-k} (-1)^m 2^m \frac{(2l_3 - 2m - 1)!!}{(2l_3 - 1)!!} \sum\nolimits_{i_1 = i_2} \left(\left\{ \left(\mathbf{x}_{l_1} \cdot (k+m) \cdot \mathbf{y}_{l_2}\right) \otimes \mathbf{I}^{\otimes m} \right\}\right)_{i_1 i_2} \\
&\quad = C_{l_1 l_2 l_3} (\qquad\qquad\qquad\qquad (A(0) + B(0)) \\
&\qquad\qquad\qquad + (-1) 2 \frac{(2l_3 - 2 - 1)!!}{(2l_3 - 1)!!} (A(1) + B(1)) \\
&\qquad\qquad\qquad\qquad \cdots \\
&\qquad\qquad + (-1)^{\tilde{m}} 2^{\tilde{m}} \frac{(2l_3 - 2\tilde{m} - 1)!!}{(2l_3 - 1)!!} (A(\tilde{m}) + B(\tilde{m})) \\
&\qquad + (-1)^{\tilde{m}+1} 2^{\tilde{m}+1} \frac{(2l_3 - 2(\tilde{m}+1) - 1)!!}{(2l_3 - 1)!!} (A(\tilde{m}+1) + B(\tilde{m}+1)) \\
&\qquad\qquad\qquad\qquad \cdots .
\end{aligned}$$

By Lemma D.3, each pair of the diagonal terms $(A(\tilde{m}), B(\tilde{m}+1))$ is cancelled taking into account an additional factor of 2^m in front of $B(\tilde{m}+1)$; $B(0)$ and $A\left(\min(l_1, l_2) - (l_1 + l_2 - l_3)/2\right)$ are left, which by definition are trivial. This result completes the proof of the traceless property for Eq. (2).

Odd irreducible Cartesian tensor product. The trace of an odd irreducible Cartesian tensor product may be written as

$$\begin{aligned}
&\mathrm{Tr}\left(\left(\mathbf{x}_{l_1} \otimes_{\mathrm{Cart}} \mathbf{y}_{l_2}\right)_{l_3}\right) \\
&\quad = \sum_{i_1 = i_2} D_{l_1 l_2 l_3} \sum\nolimits_{m=0}^{\min(l_1,l_2)-k-1} (-1)^m 2^m \frac{(2l_3 - 2m - 1)!!}{(2l_3 - 1)!!} \left(\left\{ \left(\boldsymbol{\varepsilon} : \mathbf{x}_{l_1} \cdot (k+m) \cdot \mathbf{y}_{l_2}\right) \otimes \mathbf{I}^{\otimes m} \right\}\right)_{i_1, i_2} \\
&\quad = D_{l_1 l_2 l_3} \sum\nolimits_{m=0}^{\min(l_1,l_2)-k-1} (-1)^m 2^m \frac{(2l_3 - 2m - 1)!!}{(2l_3 - 1)!!} \underbrace{\sum_{i_1 = i_2} \left(\left\{ \left(\boldsymbol{\varepsilon} : \mathbf{x}_{l_1} \cdot (k+m) \cdot \mathbf{y}_{l_2}\right) \otimes \mathbf{I}^{\otimes m} \right\}\right)_{i_1, i_2}}_{(*)_{\text{odd}}}
\end{aligned}$$

In the following, we let

$$\begin{aligned}
E_{l_3} &= \left\{i_{k_0}\right\} \subset \left\{i_1, \ldots, i_{l_3}\right\} \\
I_{l_1-(k+m)} &= \left\{i_{k_1}, \ldots, i_{k_{l_1-(k+m)}}\right\} \subset \left\{i_1, \ldots, i_{l_3}\right\} \backslash E_{l_3}, \\
J_{l_2-(k+m)} &= \left\{i_{k_{l_1-(k+m)+1}}, \ldots, i_{k_{l_3-2m}}\right\} \subset \left\{i_1, \ldots, i_{l_3}\right\} \backslash \left(I_{l_1-(k+m)} \cup E_{l_3}\right), \\
K_{l_3-2m} &= \left\{\left\{i_{k_{l_3-2m+1}}, i_{k_{l_3-2m+2}}\right\}, \ldots, \left\{i_{k_{l_3-1}}, i_{k_{l_3}}\right\}\right\}.
\end{aligned}$$

For simplicity, we omit the subscripts l_3, $l_1 - (k+m)$, $l_2 - (k+m)$, and $l_3 - 2m$ for E, I, J, and K, if there is no confusion. Then, non-trivial terms inside $(*)_{\text{odd}}$ can be written as

$$\sum_{i_1 = i_2} \left(\sum_{\substack{i_1 \in I \\ i_2 \in J}}^{(a)_m} + \sum_{\substack{i_2 \in I \\ i_1 \in J}}^{(b)_m} + \sum_{\substack{i_1 \in E \cup I \cup J \\ i_2 \in K}}^{(c)_m} + \sum_{\substack{i_2 \in E \cup I \cup J \\ i_1 \in K}}^{(d)_m} + \sum_{i_1, i_2 \in K}^{(e)_m} \right) \sum_{j_1, \ldots, j_{k+m}} \sum_{a_2, a_3} \boldsymbol{\varepsilon}_{E a_2 a_3} \left(\mathbf{x}_{l_1}\right)_{j_1 \cdots j_{k+m} a_2, I \backslash \{a_2\}} \left(\mathbf{y}_{l_2}\right)_{j_1 \cdots j_{k+m} a_3, J \backslash \{a_3\}} \mathbf{I}_K^{\otimes m}.$$

https://doi.org/10.52202/079017-3940

Note that the trace is trivial when i_1 or i_2 belongs to E and the remaining index belongs to $I \cup J$. Indeed, let $\tilde{J} = \{j_1, \ldots, j_{k+m}\}$ and suppose $i_1 \in E$ and $i_2 \in I$. Then, and we have

$$
\begin{aligned}
&\left(\operatorname{Tr}\left(\left(\mathbf{x}_{l_1} \otimes_{\text{Cart}} \mathbf{y}_{l_2}\right)_{l_3}\right)\right)_{(I\cup J)\setminus\{i_2\}} \\
&\quad= \sum_{i_1=i_2}\left(\sum_{\tilde{J}}\sum_{a_2,a_3} \boldsymbol{\varepsilon}_{i_1a_2a_3}\left(\mathbf{x}_{l_1}\right)_{\tilde{J},a_2,i_2,I\setminus\{i_2\}}\left(\mathbf{y}_{l_2}\right)_{\tilde{J},a_3,J}\right) \\
&\quad= \sum_{\tilde{J}}\left(\sum_{i_1=i_2}\sum_{a_2<a_3} \boldsymbol{\varepsilon}_{i_1a_2a_3}\left(\left(\mathbf{x}_{l_1}\right)_{\tilde{J},a_2,i_2,I\setminus\{i_2\}}\left(\mathbf{y}_{l_2}\right)_{\tilde{J},a_3,J} - \left(\mathbf{x}_{l_1}\right)_{\tilde{J},a_3,i_2,I\setminus\{i_2\}}\left(\mathbf{y}_{l_2}\right)_{\tilde{J},a_2,J}\right)\right) \\
&\quad= \sum_{\tilde{J}}\left(\sum_{\substack{(i_1,a_2,a_3)=\\(1,2,3)\\(2,1,3)\\(3,1,2)}} \boldsymbol{\varepsilon}_{i_1a_2a_3}\left(\left(\mathbf{x}_{l_1}\right)_{\tilde{J},a_2,i_1,I\setminus\{i_1\}}\left(\mathbf{y}_{l_2}\right)_{\tilde{J},a_3,J} - \left(\mathbf{x}_{l_1}\right)_{\tilde{J},a_3,i_1,I\setminus\{i_1\}}\left(\mathbf{y}_{l_2}\right)_{\tilde{J},a_2,J}\right)\right) \\
&\quad= 0.
\end{aligned}
$$

For the non-trivial terms inside $(*)_{\text{odd}}$ we have the lemma below, whose proof is identical to that of Lemma D.3.

Lemma D.4. *Let $0 \le m \le \min(l_1, l_2) - k - 2$. For each tuple $\left(E_{l_3}, I_{l_1-(k+m)}, J_{l_2-(k+m)}, K_{l_3-2m}\right)$, there exist $2l_3 - 2m - 1$ tuples of $\left(E_{l_3}, I_{l_1-(k+m+1)}, J_{l_2-(k+m+1)}, K_{l_3-2(m+1)}\right)$ such that the trace for each of the tuples equals to the trace for $\left(E_{l_3}, I_{l_1-(k+m)}, J_{l_2-(k+m)}, K_{l_3-2m}\right)$.*

Proof. Without loss of generality, we may assume

$$
\begin{aligned}
E_{l_3} &= \{i_{k_0}\} \subset \{i_1, \ldots, i_{l_3}\} \\
I_{l_1-(k+m)} &= \{i_1, i_{k_2}, \ldots, i_{k_{l_1-(k+m)}}\} \subset \{i_1, \ldots, i_{l_3}\} \setminus E_{l_3}, \\
J_{l_2-(k+m)} &= \{i_2, i_{k_{l_1-(k+m)+2}}, \ldots, i_{k_{l_3-2m}}\} \subset \{i_1, \ldots, i_{l_3}\} \setminus \left(I_{l_1-(k+m)} \cup E_{l_3}\right), \\
K_{l_3-2m} &= \{\{i_{k_{l_3-2m+1}}, i_{k_{l_3-2m+2}}\}, \ldots, \{i_{k_{l_3-1}}, i_{k_{l_3}}\}\}.
\end{aligned}
$$

Then, $\operatorname{Tr}\left((a)_m\right)$ has the following expression

$$
\begin{aligned}
&\operatorname{Tr}\left((a)_m\right) \\
&\quad= \sum_{i_1=i_2}\sum_{a_2,a_3}\sum_{j_1,\ldots,j_{k+m}} \boldsymbol{\varepsilon}_{i_{k_0}a_2a_3}\left(\mathbf{x}_{l_1}\right)_{j_1\cdots j_{k+m},a_2i_1i_{k_3}\cdots i_{k_{l_1-(k+m)}}}\left(\mathbf{y}_{l_2}\right)_{j_1\cdots j_{k+m},a_3i_2i_{k_{l_1-(k+m)+3}}\cdots i_{k_{l_3-2m}}} \mathbf{I}_K^{\otimes m} \\
&\quad= \sum_{a_2,a_3}\sum_{\substack{j_1,\ldots,j_{k+m},\\ j_{k+m+1}}} \boldsymbol{\varepsilon}_{i_{k_0}a_2a_3}\left(\mathbf{x}_{l_1}\right)_{j_1\cdots j_{k+m+1},a_2i_{k_3}\cdots i_{k_{l_1-(k+m)}}}\left(\mathbf{y}_{l_2}\right)_{j_1\cdots j_{k+m+1},a_3i_{k_{l_1-(k+m)+3}}\cdots i_{k_{l_3-2m}}} \mathbf{I}_K^{\otimes m}.
\end{aligned}
$$

Here, we assume that $a_2 \in I_{l_1-(k+m)}$ and $a_3 \in J_{l_2-(k+m)}$. The same derivation holds for $\operatorname{Tr}\left((b)_m\right)$.

For $\operatorname{Tr}((c)_{m+1})$ and $m+1$, we have $l_3 - 2m - 2$ of $\left(E_{l_3}, I_{l_1-(k+m+1)}, J_{l_2-(k+m+1)}, K_{l_3-2(m+1)}\right)$ triplets whose trace equals to $\operatorname{Tr}\left((a)_m\right)$. Indeed, like in the proof of Lemma D.2, we let

$$
\begin{aligned}
E_{l_3} \cup I_{l_1-(k+m+1)} \cup J_{l_2-(k+m+1)} &= \{i_{k_0}, i_{k_2}, \ldots, i_1, \ldots, i_{k_{l_1-(k+m)}}, i_{k_{l_1-(k+m)+2}}, \ldots, i_{k_{l_3-2m}}\}, \\
K_{l_3-2(m+1)} &= \{\{i_{k_{l_3-2m+1}}, i_{k_{l_3-2m+2}}\}, \ldots, \{i_{k_{l_3-1}}, i_{k_{l_3}}\}, \{i_{k_s}, i_2\}\}.
\end{aligned}
$$

Then, by letting $\tilde{J} = \{j_1, \ldots, j_{k+m}, j_{k+m+1}\}$, we obtain

$$
\begin{aligned}
&\mathrm{Tr}((c)_{m+1}) \\
&\quad = \sum_{i_1=i_2} \sum_{s} \sum_{a_2,a_3} \sum_{\tilde{J}} \varepsilon_{i_{k_0} a_2 a_3} \left(\mathbf{x}_{l_1}\right)_{\tilde{J} a_2 i_{k_2} \cdots i_1 \cdots i_{k_{l_1-(k+m)}}} \left(\mathbf{y}_{l_2}\right)_{\tilde{J} a_3 i_{k_{l_1-(k+m)+3}} \cdots i_{k_{l_3-2m}}} \\
&\qquad\qquad \times \delta_{i_{k_{l_3-2m+1}} i_{k_{l_3-2m+2}}} \cdots \delta_{i_{k_{l_3-1}} i_{k_{l_3}}} \delta_{i_{k_s} i_2} \\
&\quad = \sum_{s} \sum_{a_2,a_3} \sum_{\tilde{J}} \varepsilon_{i_{k_0} a_2 a_3} \left(\mathbf{x}_{l_1}\right)_{\tilde{J} a_2 i_{k_2} \cdots i_{k_s} \cdots i_{k_{l_1-(k+m)}}} \left(\mathbf{y}_{l_2}\right)_{\tilde{J} a_3 i_{k_{l_1-(k+m)+3}} \cdots i_{k_{l_3-2m}}} \\
&\qquad\qquad \times \delta_{i_{k_{l_3-2m+1}} i_{k_{l_3-2m+2}}} \cdots \delta_{i_{k_{l_3-1}} i_{k_{l_3}}} \\
&\quad = (l_3 - 2m - 2)\, \mathrm{Tr}\left((a)_m\right).
\end{aligned}
$$

The same derivation applies to $\mathrm{Tr}((d)_{m+1})$, and we get $\mathrm{Tr}((d)_{m+1}) = (l_3 - 2m - 2)\, \mathrm{Tr}\left((a)_m\right)$.

For $\mathrm{Tr}((e)_{m+1})$, the same derivation as for $\mathrm{Tr}((d)_{m+1})$ and $\mathrm{Tr}((e)_{m+1})$ in the proof of Lemma D.2 applies. Therefore, we obtain

$$\mathrm{Tr}((e)_{m+1}) = (2m+3)\, \mathrm{Tr}\left((a)_m\right).$$

Finally, we can write

$$
\begin{aligned}
&\mathrm{Tr}((c)_{m+1}) + \mathrm{Tr}((d)_{m+1}) + \mathrm{Tr}((e)_{m+1}) \\
&\quad = (l_3 - 2m - 2)\, \mathrm{Tr}((a)_m) + (l_3 - 2m - 2)\, \mathrm{Tr}((a)_m) + (2m+3)\, \mathrm{Tr}((a)_m) \\
&\quad = (2l_3 - 2m - 1)\, \mathrm{Tr}((a)_m) = (2l_3 - 2m - 1)\, \mathrm{Tr}((b)_m),
\end{aligned}
$$

which completes the proof of the Lemma. □

Noting that $\mathrm{Tr}\left((a)_m\right)$ is doubled for every m due to the factor 2^m, the proof is completed similarly to the proof of Lemma D.3. This result completes the proof of the traceless property for Eq. (3).

E Experiments and results

E.1 Description of the data sets

rMD17 data set. The revised MD17 (rMD17) data set is a collection of structures, energies, and atomic forces of ten small organic molecules obtained from ab initio molecular dynamics (AIMD) [47]. These molecules are derived from the original MD17 data set [47, 84–86], with 100,000 structures sampled for each. Our models are trained using 950 and 50 configurations for each molecule randomly sampled from the original data set using five random seeds, with 50 additional configurations randomly sampled for early stopping. We use the remaining configurations to test the final models. Table 1 reports the mean absolute errors (MAEs) in total energies and atomic forces averaged over five independent runs, including the standard deviation between them.

MD22 data set. The MD22 data set includes structures, energies, and atomic forces of seven molecular systems derived from AIMD simulations at elevated temperatures, spanning four major classes of biomolecules and supramolecules [48]. These molecular systems range from a small peptide containing 42 atoms to a double-walled nanotube comprising 370 atoms. Characterized by complex intermolecular interactions, this data set was designed to challenge short-range models. The training set sizes used in this study are consistent with those in the original publication [48], selected to ensure that the sGDML model stays within a target accuracy of 1 kcal/mol (≈ 43.37 meV). We randomly selected an additional set of 500 structures for each molecule in the data set for early stopping, while the remaining configurations were reserved for testing the final models. The corresponding training and validation data sets were randomly selected using three random seeds. Table A2 reports MAEs in total energies per atom and atomic forces averaged over three independent runs, including the standard deviation between them.

3BPA data set. The 3BPA data set comprises structures, energies, and atomic forces of a flexible drug-like organic molecule obtained from AIMD at various temperatures [49]. The training data set consists of 500 configurations sampled at 300 K, while three separate test data sets are obtained from

 https://doi.org/10.52202/079017-3940

AIMD simulations at 300 K, 600 K, and 1200 K. An additional test data set provides energy values along dihedral rotations of the molecule. This test directly assesses the smoothness and accuracy of the potential energy surface, influencing properties such as binding free energies to protein targets. Our models are trained using 450 and 50 configurations randomly sampled from the training data set using five random seeds, with further 50 configurations reserved for early stopping. Table 2 reports the root-mean-square errors (RMSEs) in total energies and atomic forces averaged over five independent runs for a training set size of 450 structures, including the standard deviation between them. Table A3 presents the corresponding results obtained with a training set size of 50 structures.

Acetylacetone data set. The acetylacetone data set includes structures, energies, and atomic forces of a small reactive molecule obtained from AIMD at various temperatures [32]. The training data set comprises configurations sampled at 300 K, while the test data sets are sampled at 300 K and 600 K. The generalization ability of final models is evaluated using an elevated temperature of 600 K and along two internal coordinates of the molecule: The hydrogen transfer path and a partially conjugated double bond rotation featuring a high rotation barrier. Our models are trained using 450 and 50 configurations randomly sampled from the training dataset using five random seeds, with further 50 configurations reserved for early stopping. Table 3 reports RMSEs in total energies and atomic forces averaged over five independent runs for a training set size of 450 structures, including the standard deviation between them. Table A5 presents the corresponding results obtained with a training set size of 50 structures.

Ta–V–Cr–W data set. The Ta–V–Cr–W data set includes 0 K energies, atomic forces, and stresses for binaries, ternaries, and quaternary and near-melting temperature properties in four-component disordered high-entropy alloys [50]. In total, this benchmark data set comprises 6711 configurations, with energies, atomic forces, and stresses computed at the density functional theory (DFT) level. More precisely, there are 5680 0 K structures: 4491 binary, 595 ternary, and 594 quaternary structures, along with 1031 structures sampled from MD at 2500 K. Structure sizes range from 2 to 432 atoms in the periodic cell. All models are trained using 5373 configurations, with 4873 used for training and 500 for early stopping. The remaining configurations are reserved for testing the models' performance. The performance is evaluated separately using 0 K binaries, ternaries, quaternaries, and near-melting temperature four-component disordered alloys. The original data set provides ten different training-test splits. In our experiments, each training set is further split into training and validation subsets using a random seed. Furthermore, we use additional binary structures strained along the $[100]$ direction as a part of the test data set. Note that the final models, in this case, were obtained by training using the whole data set of 6711 configurations (training + test), 500 of which were reserved as a validation data set. Table A6 reports RMSEs in total energies per atom and atomic forces averaged over ten independent runs, including the standard deviation between them.

E.2 Training details

All ICTP and MACE models employed in this work were trained on a single NVIDIA A100 GPU with 80 GB of RAM. Training times for ICTP and MACE models typically ranged from 30 minutes to a few days, depending on the data set, the data set size, and the employed precision. We used double precision for 3BPA and acetylacetone data sets and single precision for rMD17, in line with the original experiments [33]. Double precision was also used for MD22, while single precision was employed in our Ta–V–Cr–W experiments. Unless stated otherwise, we used two message-passing layers and irreducible Cartesian tensors or spherical tensors of a maximal rank of $l_{\max} = 3$ to embed the directional information of atomic distance vectors.

For ICTP models with the full ($\text{ICTP}_{\text{full}}$) and symmetric ($\text{ICTP}_{\text{sym}}$) product basis and MACE, we employ 256 uncoupled feature channels. Exceptions include our experiments with the 3BPA data set, aimed at investigating scaling and computational cost, and the Ta–V–Cr–W experiments, where we used eight and 32 feature channels, respectively. For ICTP models with the symmetric product basis evaluated in the latent feature space ($\text{ICTP}_{\text{sym+lt}}$), we use 64 coupled feature channels for the Cartesian product basis and 256 for two-body features. Radial features are derived from eight Bessel basis functions with polynomial envelope for the cutoff with $p = 5$ [60]. These features are fed into a fully connected NN of size $[64, 64, 64]$. We apply SiLU non-linearities to the outputs of the hidden layers [87, 88]. The readout function of the first message-passing layer is implemented as a linear layer. The readout function of the second layer is a single-layer fully connected NN with 16 hidden

neurons. A cutoff radius of 5.0 Å is used across all data sets except MD22, where we used a cutoff radius of 5.5 Å for the double-walled nanotube and 6.0 Å for the other molecules in the data set.

All parameters of ICTP and MACE models were optimized by minimizing the combined squared loss on training data $\mathcal{D}_{\text{train}} = (\mathcal{X}_{\text{train}}, \mathcal{Y}_{\text{train}})$, where $\mathcal{X}_{\text{train}} = \{\mathcal{S}^{(k)}\}_{k=1}^{N_{\text{train}}}$ and $\mathcal{Y}_{\text{train}} = \{E_k^{\text{ref}}, \{\mathbf{F}_{u,k}^{\text{ref}}\}_{u=1}^{N_{\text{at}}}, \boldsymbol{\sigma}_k^{\text{ref}}\}_{k=1}^{N_{\text{train}}}$

$$\mathcal{L}\left(\boldsymbol{\theta}, \mathcal{D}_{\text{train}}\right) = \sum_{k=1}^{N_{\text{train}}} \left[C_{\text{e}} \left\| E_k^{\text{ref}} - E(S^{(k)}, \boldsymbol{\theta}) \right\|_2^2 + C_{\text{f}} \sum_{u=1}^{N_{\text{at}}^{(k)}} \left\| \mathbf{F}_{u,k}^{\text{ref}} - \mathbf{F}_u\left(S^{(k)}, \boldsymbol{\theta}\right) \right\|_2^2 + C_{\text{s}} \left\| V_k \boldsymbol{\sigma}_k^{\text{ref}} - V_k \boldsymbol{\sigma}\left(S^{(k)}, \boldsymbol{\theta}\right) \right\|_2^2 \right]. \tag{A5}$$

Here, $\boldsymbol{\sigma}_k^{\text{ref}}$ is the stress tensor defined as $\boldsymbol{\sigma} = \frac{1}{V} \nabla_{\boldsymbol{\epsilon}} E|_{\boldsymbol{\epsilon}=\mathbf{0}}$, where E denotes total energy after a strain deformation with symmetric tensor $\boldsymbol{\epsilon} \in \mathbb{R}^{3\times3}$ and V is the volume of the periodic box.

When training ICTP models on rMD17, 3BPA, and acetylacetone data sets, we neglected the stress loss and set $C_{\text{e}} = 1/N_{\text{at}}^{(k)}$ and $C_{\text{f}} = 10$ Å^2 to balance the relative contributions of total energies and atomic forces, respectively. For MACE and 3BPA/acetylacetone, $C_{\text{e}} = 1/(B \times N_{\text{at}}^{(k)})$ and $C_{\text{f}} = 1000/(B \times 3 \times N_{\text{at}}^{(k)})$ Å^2 were used with B denoting the batch size. For ICTP models trained on the MD22 data set, we set $C_{\text{e}} = 10/N_{\text{at}}^{(k)}$ and $C_{\text{f}} = 1$ Å^2, using energies and forces in eV and eV/Å, respectively. For the Ta–V–Cr–W dataset, the stress loss was incorporated into the combined loss in Eq. (A5), along with the energy and force losses. For ICTP, we used $C_{\text{e}} = 1/N_{\text{at}}^{(k)}$, $C_{\text{f}} = 0.01$ Å^2, and $C_{\text{s}} = 0.001/N_{\text{at}}^{(k)}$ to balance the relative contributions of total energies, atomic forces, and stresses, respectively. For MACE, we chose $C_{\text{e}} = 1/(B \times N_{\text{at}}^{(k)} \times N_{\text{at}}^{(k)})$, $C_{\text{f}} = 1/(B \times 3 \times N_{\text{at}}^{(k)})$ Å^2, and $C_{\text{s}} = 0.05/(B \times 9 \times N_{\text{at}}^{(k)} \times N_{\text{at}}^{(k)})$. Here, $E(\mathcal{S}^{(k)}, \boldsymbol{\theta})$, $\mathbf{F}_u\left(\mathcal{S}^{(k)}, \boldsymbol{\theta}\right)$, and $\boldsymbol{\sigma}\left(S^{(k)}, \boldsymbol{\theta}\right)$ are total energies, atomic forces, and stresses predicted by ICTP or MACE.

All models for rMD17, 3BPA, and acetylacetone were trained for 2000 epochs using the AMSGrad variant of Adam [89], with default parameters of $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-8}$. For MD22 and Ta–V–Cr–W, all models were trained for 1000 epochs. For rMD17, 3BPA, and acetylacetone data sets, we used a learning rate of 0.01 and a batch size of 5. For MD22 and Ta–V–Cr–W, we again chose a learning rate of 0.01 but a mini-batch of 2 and 32, respectively. For evaluations on the validation and test data sets, we used a batch size of 10 for rMD17, 3BPA, and acetylacetone, while mini-batches of 2 and 32 were used for MD22 and Ta–V–Cr–W, respectively.

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 for all data sets except for MD22, for which we used a patience of 10. We utilize an exponential moving average with a weight of 0.99 for evaluation on the validation set and for the final model. Additionally, in line with MACE [33], we apply exponential weight decay of 5×10^{-7} on the weights of Eqs. (6), (A1), and (A3). Furthermore, we incorporate a per-atom shift of total energies via the average per-atom energy over all the training configurations, including the energies of individual atoms for 3BPA and acetylacetone datasets. If no atomic energies are provided, as for rMD17, MD22, and Ta–V–Cr–W, the per-atom shift is obtained by solving a linear regression problem [67]. Additionally, a per-atom scale is determined as the root-mean-square of the components of the forces over the training configurations.

E.3 Additional results

Scaling and computational cost. Table A1 provides the numerical results complementing Fig. 2 in the main text. All results in Table A1 and Fig. 2 were obtained using irreducible Cartesian tensors with a maximal rank of $l_{\max} = L$ to represent local atomic environments. Here, L denotes the tensor rank of employed equivariant messages. We facilitated the exploration of larger ν values by setting the number of uncoupled feature channels to eight. In the MACE model, intermediate spherical tensors with a rank of $l > l_{\max}$ are used to construct the product basis. However, the pre-computation of generalized Clebsch–Gordan coefficients for $\nu > 4$, in some cases, would require more than 2 TB of RAM. Therefore, in our experiments, we fixed the maximum rank of intermediate tensors to $l = l_{\max}$. We also used the full product basis for ICTP to calculate the same number of ν-fold tensor

 https://doi.org/10.52202/079017-3940

Table A1: **Inference times and memory consumption as a function of the tensor rank L and the correlation order ν for the 3BPA data set.** All values for ICTP and MACE models are obtained by averaging over five independent runs. The standard deviation is provided if it is available. Best performances are highlighted in bold. Inference time and memory consumption are measured for a batch size of 10. Inference time is reported per structure in ms; memory consumption is provided for the entire batch in GB.

	$L=1$		$L=2$		$L=3$	
	$\text{ICTP}_{\text{full}}$	MACE	$\text{ICTP}_{\text{full}}$	MACE	$\text{ICTP}_{\text{full}}$	MACE
			Inference times			
$\nu=1$	$\mathbf{0.76 \pm 0.17}$	1.02 ± 0.03	$\mathbf{0.87 \pm 0.18}$	1.38 ± 0.04	$\mathbf{0.98 \pm 0.26}$	1.88 ± 0.03
$\nu=2$	$\mathbf{0.59 \pm 0.20}$	1.12 ± 0.03	$\mathbf{1.03 \pm 0.21}$	1.52 ± 0.05	$\mathbf{1.34 \pm 0.08}$	2.0 ± 0.10
$\nu=3$	$\mathbf{0.79 \pm 0.22}$	1.23 ± 0.03	$\mathbf{1.15 \pm 0.08}$	1.67 ± 0.03	$\mathbf{1.85 \pm 0.13}$	2.23 ± 0.03
$\nu=4$	$\mathbf{0.94 \pm 0.17}$	1.41 ± 0.11	$\mathbf{1.31 \pm 0.21}$	1.83 ± 0.01	$\mathbf{2.07 \pm 0.20}$	2.53 ± 0.01
$\nu=5$	$\mathbf{1.02 \pm 0.17}$	1.52 ± 0.08	$\mathbf{1.72 \pm 0.07}$	2.26 ± 0.03	$\mathbf{3.61 \pm 0.02}$	OOM
$\nu=6$	$\mathbf{1.0 \pm 0.07}$	1.77 ± 0.05	$\mathbf{1.83 \pm 0.16}$	27.85 ± 0.01	$\mathbf{16.76 \pm 0.35}$	OOM
			Memory consumption			
$\nu=1$	0.05 ± 0.00	$\mathbf{0.04 \pm 0.00}$	0.08 ± 0.00	$\mathbf{0.06 \pm 0.00}$	0.21 ± 0.00	$\mathbf{0.13 \pm 0.00}$
$\nu=2$	0.05 ± 0.00	$\mathbf{0.04 \pm 0.00}$	0.08 ± 0.00	$\mathbf{0.07 \pm 0.00}$	0.28 ± 0.09	$\mathbf{0.13 \pm 0.00}$
$\nu=3$	0.05 ± 0.00	$\mathbf{0.04 \pm 0.00}$	0.10 ± 0.00	$\mathbf{0.08 \pm 0.00}$	0.51 ± 0.03	$\mathbf{0.23 \pm 0.00}$
$\nu=4$	$\mathbf{0.05 \pm 0.00}$	$\mathbf{0.05 \pm 0.00}$	$\mathbf{0.18 \pm 0.08}$	0.30 ± 0.00	$\mathbf{1.07 \pm 0.10}$	4.16 ± 0.00
$\nu=5$	$\mathbf{0.05 \pm 0.00}$	0.07 ± 0.00	$\mathbf{0.35 \pm 0.07}$	3.18 ± 0.00	$\mathbf{5.07 \pm 0.02}$	OOM
$\nu=6$	$\mathbf{0.11 \pm 0.09}$	0.22 ± 0.00	$\mathbf{0.93 \pm 0.00}$	50.49 ± 0.00	$\mathbf{28.48 \pm 0.03}$	OOM

Table A2: **Energy (E) and force (F) mean absolute errors (MAEs) for the MD22 data set.**[a] E- and F-MAE are given in meV/atom and meV/Å, respectively. Results are shown for models trained using training set sizes defined in the original publication [48]. All values for ICTP models are obtained by averaging over three independent runs. We also use an additional subset of 500 configurations drawn randomly from the original data set for early stopping. The standard deviation is provided if available. Best performances, considering the standard deviation, are highlighted in bold.

		ICTP_{sym}	ViSNet-LSRM [90]	Equiformer [90]	So3krates [91]	MACE [61]	Allegro [90]	TorchMD-Net [90]	sGDML [48]
Ac–Ala3–NHMe	E	0.068 ± 0.000	0.068	0.085	0.348	**0.064**	0.105	0.116	0.403
	F	$\mathbf{3.28 \pm 0.02}$	3.91	3.49	10.58	3.8	4.63	8.15	34.26
DHA	E	0.080 ± 0.001	**0.068**	0.138	0.293	0.102	0.089	0.093	0.997
	F	2.62 ± 0.01	2.59	**2.19**	10.49	2.8	3.17	5.24	32.52
Stachyose	E	$\mathbf{0.053 \pm 0.001}$	**0.053**	0.070	0.22	0.062	0.124	0.069	1.995
	F	$\mathbf{2.51 \pm 0.03}$	3.33	2.75	18.86	3.8	4.21	8.33	29.49
AT–AT	E	$\mathbf{0.057 \pm 0.002}$	**0.056**	0.095	0.129	0.079	0.103	0.081	0.52
	F	$\mathbf{3.02 \pm 0.07}$	3.39	4.16	9.37	4.3	4.13	8.83	29.92
AT–AT–CG–CG	E	0.045 ± 0.001	**0.042**	0.055	0.127	0.058	0.145	0.076	0.52
	F	$\mathbf{3.37 \pm 0.05}$	4.61	5.43	14.40	5.0	5.55	14.13	30.36
Buckyball catcher	E	0.123 ± 0.001	0.124	0.117	**0.112**	0.141	0.154	0.152	0.343
	F	$\mathbf{3.58 \pm 0.07}$	4.45	4.83	10.28	3.7	3.85	14.39	29.49
Double-walled nanotube	E	0.352 ± 0.003[b]	0.214	0.140	**0.116**	0.194	0.259	0.173	0.468
	F	12.64 ± 0.23[b]	14.71	**11.91**	31.53	12.0	14.87	43.5	22.55

[a] ICTP, MACE, Equiformer, Allegro, and TorchMD-Net are short-range models (i.e., they use local or semi-local atomic representations), ViSNet-LSRM and SO3krates include long-range information, and sGDML is a global model.

[b] For consistency, these results are obtained for relative energy and force weights of $C_{\text{e}} = 10/N_{\text{at}}^{(k)}$ and $C_{\text{f}} = 1$ Å^2, based on our experiments with the Ac–Ala3–NHMe molecule. For comparison, we also trained ICTP for 800 epochs with $C_{\text{e}} = 100/N_{\text{at}}^{(k)}$ and $C_{\text{f}} = 1$ Å^2, achieving MAEs of 0.209 ± 0.007 meV/atom for energy and 13.98 ± 0.30 meV/Å for force.

products, i.e., $\mathcal{K} = \text{len}\,(\eta_\nu)$, as used in MACE with $l = l_{\max}$. Finally, we note that ICTP and MACE use different approaches to optimize their runtimes; however, the scaling with respect to the tensor rank and the correlation order is independent of these optimization methods.

Table A3: **Energy (E) and force (F) root-mean-square errors (RMSEs) for the 3BPA data set (results for $N_{\mathrm{train}} = 50$).** E- and F-RMSE are given in meV and meV/Å, respectively. Results are shown for models trained using 50 molecules randomly drawn from the training data set collected at 300 K, with further 50 used for early stopping. All ICTP and MACE results are obtained by averaging over five independent runs, with the standard deviation provided if available. Best performances, considering the standard deviation, are highlighted in bold.

		$\mathrm{ICTP_{full}}$	$\mathrm{ICTP_{sym}}$	$\mathrm{ICTP_{sym+lt}}$	MACE
300 K	E	**14.98 ± 1.62**	**13.43 ± 1.00**	**16.03 ± 1.26**	**14.54 ± 1.02**
	F	**37.21 ± 2.14**	**37.27 ± 1.63**	**38.38 ± 1.55**	**37.68 ± 1.33**
600 K	E	**31.68 ± 3.56**	**31.63 ± 3.94**	**30.74 ± 1.82**	**30.71 ± 3.43**
	F	**69.87 ± 3.04**	**68.87 ± 2.94**	**69.62 ± 2.88**	**69.88 ± 3.88**
1200 K	E	**92.16 ± 9.33**	**86.0 ± 12.03**	**78.51 ± 8.88**	**83.99 ± 8.20**
	F	**157.72 ± 6.53**	**153.16 ± 9.62**	**151.37 ± 9.59**	**154.46 ± 10.84**
Dihedral slices	E	**33.69 ± 8.03**	**31.06 ± 6.27**	**33.44 ± 5.56**	**27.79 ± 7.41**
	F	**47.12 ± 5.27**	**47.68 ± 2.21**	**49.08 ± 4.05**	**48.91 ± 4.71**

Molecular dynamics trajectories. Table A2 presents the energy and force mean absolute errors (MAEs) for the ICTP models trained on the MD22 data set. This data set was designed to challenge short-range potential models—i.e., those based on local or semi-local atomic representations—and includes large molecular systems with complex intermolecular interactions. We compare errors obtained for ICTP models with those of state-of-the-art approaches that incorporate global, short- and long-range information. From Table A2, we see that ICTP achieves accuracy in predicted energies and atomic forces on par with or better than state-of-the-art methods. The ICTP model uses a cutoff of 6.0 Å (5.5 Å for the double-walled nanotube), resulting in a receptive field of 12.0 Å (11.0 Å), considering the two message-passing layers. This receptive field is, in most cases, smaller than the diameter of molecular systems in the data set. Thus, ICTP is, at most, a semi-local potential model—similar to MACE, which used a 5.0 Å cutoff and two message-passing layers.

We chose a larger cutoff for ICTP than the one used by MACE motivated by the results in the recent work [7]. In particular, the authors compared MACE to VisNet-LSRM—the best model reported to date for MD22, which employs mixed short- and long-range information in their message passing—and reported that MACE can achieve lower force errors. However, VisNet-LSRM typically had lower energy errors, attributed to the improvement from considering atomic interactions beyond 10.0 Å. Using a larger cutoff radius for ICTP compared to MACE, we expected results closer to those of VisNet-LSRM. Table A2 shows that using a receptive field of 12.0 Å is, in most cases, sufficient to achieve accuracy in predicted energies close to the one obtained by VisNet-LSRM.

Extrapolation to out-of-domain data. Table A3 demonstrates total energy and atomic force RMSEs obtained for ICTP and MACE models trained using 50 configurations randomly drawn from the original data set. ICTP and MACE models perform similarly, considering the standard deviation obtained across five independent runs. However, ICTP models often have lower mean RMSE values compared to MACE. Furthermore, Table A4 presents the total energy and atomic force RMSEs for ICTP and MACE models that use $\nu = 1$ to compute the product basis. Thus, we provide results for models which rely exclusively on two-body interactions. We note that for $\nu = 1$, $\mathrm{ICTP_{full}}$ and $\mathrm{ICTP_{sym}}$ are identical; though, we include both results for completeness.

Figure A1 compares potential energy profiles obtained with ICTP and MACE models trained using 450 configurations. Potential energy cuts at $\beta = 120°$ and $\beta = 180°$ are easier tasks for MLIPs, as there are training points in the data set with similar combinations of dihedral angles [32]. In contrast, the potential energy cut at $\beta = 150°$ is more challenging, with no training points close to it. Notably, Fig. A1 shows that all models produce smooth potential energy profiles close to the reference ones (DFT) for all values of β. These results again demonstrate excellent extrapolation capabilities of irreducible Cartesian models that are on par with the spherical MACE model.

Flexibility and reactivity. Table A5 demonstrates total energy and atomic force RMSEs obtained for ICTP and MACE models trained using 50 configurations randomly drawn from the original data set.

 https://doi.org/10.52202/079017-3940

Table A4: **Energy (E) and force (F) root-mean-square errors (RMSEs) for the 3BPA data set (results for $\nu = 1$).** E- and F-RMSE are given in meV and meV/Å, respectively. Results are shown for models trained using 450 configurations randomly drawn from the training data set collected at 300 K, with further 50 used for early stopping. For all models, $\nu = 1$ is used. Thus, the employed models rely exclusively on two-body interactions. All ICTP and MACE results are obtained by averaging over five independent runs. Best performances, considering the standard deviation, are highlighted in bold. Inference time and memory consumption are measured for a batch size of 100. Inference time is reported per structure in ms, while memory consumption is provided for the entire batch in GB.

		$\text{ICTP}_{\text{full}}$	ICTP_{sym}	$\text{ICTP}_{\text{sym+lt}}$	MACE
300 K	E	**12.90 ± 1.06**	**12.90 ± 1.06**	**14.97 ± 1.64**	**13.50 ± 1.71**
	F	**29.90 ± 0.25**	**29.90 ± 0.25**	30.93 ± 0.47	**30.18 ± 0.38**
600 K	E	**29.97 ± 0.94**	**29.97 ± 0.94**	**31.64 ± 1.83**	**31.32 ± 2.16**
	F	**62.80 ± 0.45**	**62.80 ± 0.45**	64.54 ± 0.95	**63.04 ± 0.73**
1200 K	E	**81.03 ± 1.64**	**81.03 ± 1.64**	**79.26 ± 4.66**	**81.54 ± 2.02**
	F	**146.96 ± 1.30**	**146.96 ± 1.30**	151.45 ± 5.21	149.44 ± 1.94
Dihedral slices	E	**22.84 ± 2.96**	**22.84 ± 2.96**	29.16 ± 7.96	28.08 ± 4.04
	F	**48.82 ± 5.25**	**48.82 ± 5.25**	**52.53 ± 5.27**	**49.62 ± 2.92**
Inference time		2.63 ± 0.02	2.62 ± 0.02	**2.53 ± 0.01**	2.96 ± 0.06
Memory consumption		32.57 ± 0.00	32.57 ± 0.00	32.34 ± 0.09	**23.32 ± 0.00**

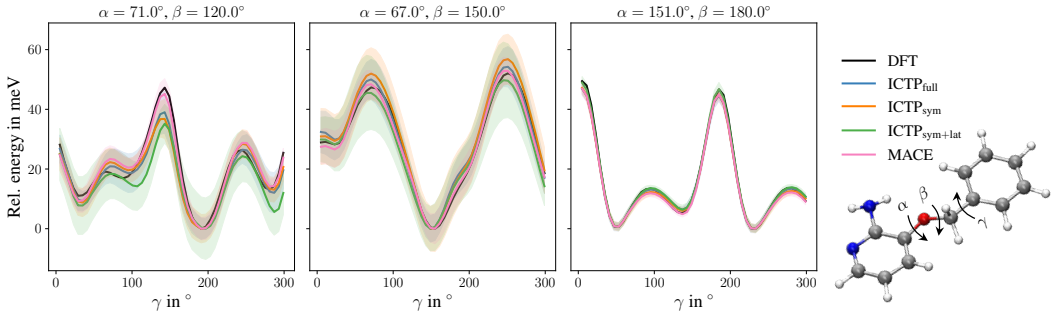

Figure A1: **Potential energy profiles for three cuts through the 3BPA molecule's potential energy surface (results for $N_{\text{train}} = 450$).** All models are trained using 450 configurations, and the remaining 50 are used for early stopping. The 3BPA molecule, including the three dihedral angles (α, β, and γ), provided in degrees °, is shown as an inset. The color code of the inset molecule is C grey, O red, N blue, and H white. The reference potential energy profile (DFT) is shown in black. Each profile is shifted such that each model's lowest energy is zero. Shaded areas denote standard deviations across five independent runs.

Similar to the 3BPA data set, ICTP and MACE models demonstrate comparable accuracy in predicted energies and forces, considering the standard deviation obtained across five independent runs.

Figure A2 further investigates the generalization capabilities of ICTP models trained using 450 configurations, demonstrating potential energy profiles for the rotation around the C-C bond, i.e., the C-C-C-O dihedral angle (α), and for the hydrogen transfer (i.e., the O–H distance d_{OH} in Fig. A2). The training data set encompasses dihedral angles less than 30°. Furthermore, the energy barrier of 1 eV is outside the energy range of the training data set obtained at 300 K. As for the hydrogen transfer, the training data does not contain transition geometries, but the reaction still occurs in a region that is not too far from the training data. Overall, ICTP models perform on par with MACE for predicting potential energy profiles for the rotation around the corresponding C-C bond and for the hydrogen transfer.

Table A5: **Energy (E) and force (F) root-mean-square errors (RMSEs) for the acetylacetone data set (results for $N_{\text{train}} = 50$).** E- and F-RMSE are given in meV and meV/Å, respectively. Results are shown for models trained using 50 configurations randomly drawn from the training data set collected at 300 K, with further 50 used for early stopping. All ICTP and MACE results are obtained by averaging over five independent runs. Best performances, considering the standard deviation, are highlighted in bold.

		$\text{ICTP}_{\text{full}}$	ICTP_{sym}	$\text{ICTP}_{\text{sym+lt}}$	MACE
300 K	E	**4.42 ± 0.39**	**4.45 ± 0.36**	**4.38 ± 0.24**	**4.22 ± 0.52**
	F	**28.85 ± 3.00**	**28.28 ± 1.45**	**29.17 ± 1.65**	**28.38 ± 2.74**
600 K	E	**17.61 ± 3.14**	**16.13 ± 1.37**	**17.3 ± 2.05**	**17.53 ± 3.58**
	F	**75.09 ± 8.70**	**69.99 ± 5.12**	**74.96 ± 4.68**	**74.93 ± 9.91**

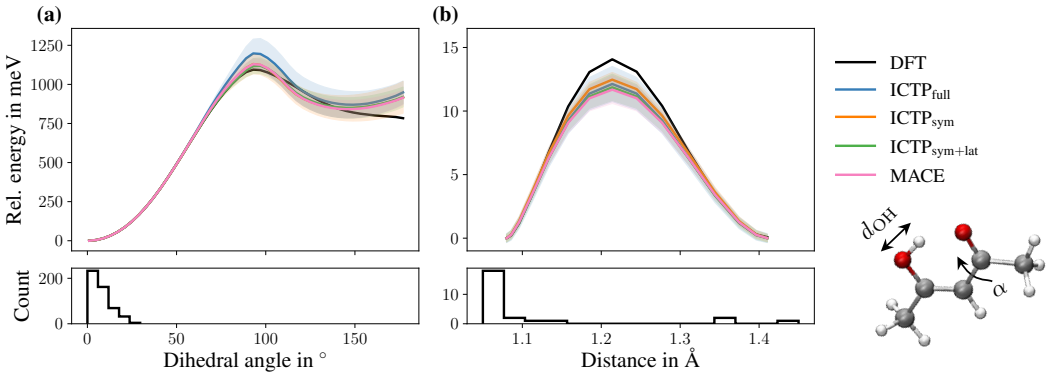

Figure A2: **Potential energy profiles of (a) the dihedral angle describing the rotation around the C-C bond and (b) hydrogen transfer between two oxygen atoms (results for $N_{\text{train}} = 450$).** All models are trained using 450 molecules, and the remaining 50 are used for early stopping. The acetylacetone molecule, including the dihedral angle in degrees ° describing the rotation around the C-C bond (α), is shown as an inset in (a). The color code of the inset molecule is C grey, O red, and H white. The reference potential energy profile (DFT) is shown in black. Each profile is shifted such that each model's lowest energy is zero. The histograms demonstrate the distribution of dihedral angles and O-H distances in the training data. Shaded areas denote standard deviations across five independent runs.

Figure A3 shows potential energy profiles for MLIPs trained with 50 configurations, similar to Fig. A2. Notably, Fig. A2 (b) demonstrates that $\text{ICTP}_{\text{full}}$ is the only MLIP consistently producing the potential energy profile for the hydrogen transfer close to the reference (DFT). This task is particularly challenging, as most data splits do not include configurations sufficiently close to the transition structure as in Fig. A2.

Multicomponent alloys. The Ta–V–Cr–W data set is designed to evaluate the performance of MLIPs across atomic systems with varying numbers of atom types/components, comprising both relaxed (0 K) and high-temperature structures [50]. In particular, this data set includes 0 K energies, forces, and stresses for 2-, 3-, and 4-component systems and 2500 K properties in 4-component disordered alloys. It contains 6711 configurations with sizes ranging from two to 432 atoms in the periodic cell. Table A6 demonstrates the energy and force RMSEs for the ICTP and MACE models, evaluated separately on 0 K binaries, ternaries, quaternaries, and near-melting temperature four-component disordered alloys. ICTP consistently outperforms state-of-the-art models for the Ta–V–Cr–W data set, i.e., MTP and GM-NN. In contrast, for MACE, we were not able to identify a set of relative weights for energy, forces, and virial losses that consistently yield results better than those of MTP and GM-NN in both energies and forces. Table A6 shows that MACE often matches the accuracy of ICTP on forces but is typically outperformed by a factor of ≤ 2.0 on energies.

https://doi.org/10.52202/079017-3940

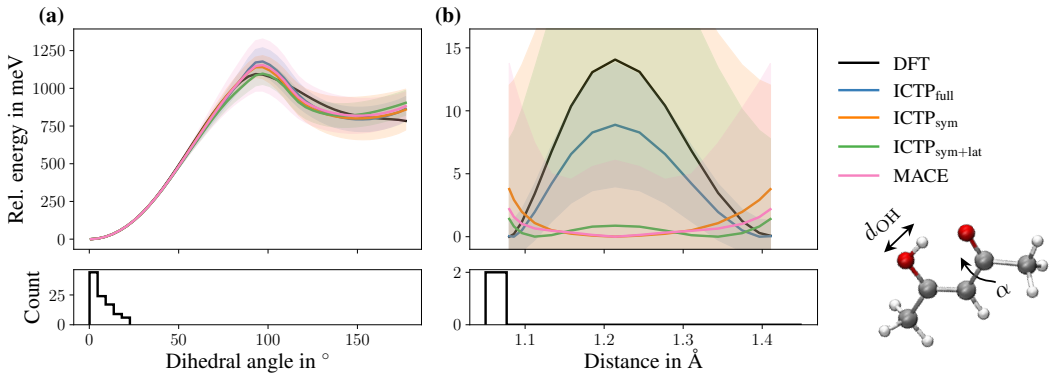

Figure A3: **Potential energy profiles of (a) the dihedral angle describing the rotation around the C-C bond and (b) hydrogen transfer between two oxygen atoms (results for $N_{\text{train}} = 50$).** All models are trained using 50 molecules, and additional 50 are used for early stopping. The acetylacetone molecule, including the dihedral angle in degrees ° describing the rotation around the C-C bond (α), is shown as an inset in (a). The color code of the inset molecule is C grey, O red, and H white. The reference potential energy profile (DFT) is shown in black. Each profile is shifted such that each model's lowest energy is zero. The histograms demonstrate the distribution of dihedral angles and O-H distances in the training data set for one random seed used to split the original data. Shaded areas denote standard deviations across five independent runs.

We further compare the ICTP, MACE, MTP, and GM-NN models using the separate test data set containing binary structures strained along the [100] direction. We find that neither ICTP nor MACE consistently outperforms MTP and GM-NN in this case. However, because this test data set contains only a single configuration of 432 atoms per binary, it may not serve as a valuable benchmark in this study and is included merely for completeness. Additionally, we trained a Cartesian model with $\nu = 2$ and $L = l_{\text{max}} = 2$, which resembles a TensorNet-like architecture [30], though it incorporates equivariant convolution filters. The corresponding results are provided in Table A6. We found that model configurations with $\nu = 3$ and $L = 2$ (and often with $L = 1$) outperform the $\nu = 2$ and $L = l_{\text{max}} = 2$ configuration by factors of ≤ 1.4 and ≤ 1.2 in energy and force RMSEs, respectively.

Finally, similar to our results for the 3BPA data set, we observe that MACE can be more computationally efficient than ICTP. We attribute longer inference times of ICTP to the pre-factor $\mathcal{K}$ arising from the Cartesian product basis in Eq. (6). Thus, we attribute ICTP's lower computational efficiency to the use of the MACE architecture, which is optimized for spherical tensors. We expect that further modifications to the architecture can facilitate more efficient use of the advantages of operations based on irreducible Cartesian tensors.

F Broader social impact

This section discusses the broader social impact of the presented work. Our work has important implications for the chemical sciences and engineering, as many problems in these fields require atomistic simulations; we also discuss it in Section 1. Although this work focuses on standard benchmark data sets, our experiments demonstrate the scalability of our method to larger atomic systems. Beyond constructing machine-learned interatomic potentials, equivariant models based on irreducible Cartesian tensors can be applied for molecular property prediction, protein structure prediction, protein generation, ribonucleic acid structure ranking, and many more.

Our work has no obvious negative social impact. As long as it is applied to the chemical sciences and engineering in a way that benefits society, it will have positive effects.

Table A6: **Energy (E) and force (F) root-mean-square errors (RMSEs) for the Ta–V–Cr–W data set.** E- and F-RMSEs are given in meV/atom and eV/Å. Results are obtained by averaging over ten splits of the original data set, except for the deformed structures. For the latter, the results are obtained using the whole data set (training + test). For the ICTP, MACE, and GM-NN models, we randomly selected a validation data set of 500 structures from the corresponding training data sets. Best performances, considering the standard deviation, are highlighted in bold. Inference time and memory consumption are measured for a batch size of 50. Inference time[a] is reported per atom in μs; memory consumption is provided for the entire batch in GB.

Subsystem		ICTP$_{\text{sym}}$ ($L = 2$)	ICTP$_{\text{sym}}$ ($L = 1$)	ICTP$_{\text{sym}}$ ($L = 0$)	MACE ($L = 2$)	MACE ($L = 1$)	MACE ($L = 0$)	ICTP$_{\text{sym}}$ ($\nu = 2$)	MTP [50]	GM-NN [50]	EAM [50]
TaV	E	**1.02 ± 0.27**	**1.21 ± 0.54**	1.65 ± 1.06	1.72 ± 0.67	1.76 ± 0.53	2.24 ± 1.34	**1.24 ± 0.50**	1.94	1.54	32.00
	F	**0.020 ± 0.002**	0.022 ± 0.002	0.024 ± 0.002	**0.019 ± 0.002**	**0.020 ± 0.003**	0.022 ± 0.002	0.023 ± 0.002	0.050	0.029	0.404
TaCr	E	**1.81 ± 0.29**	**1.94 ± 0.23**	2.13 ± 0.19	3.26 ± 0.42	3.31 ± 0.44	4.18 ± 0.56	2.40 ± 0.33	3.26	2.98	43.60
	F	**0.025 ± 0.007**	**0.024 ± 0.006**	0.027 ± 0.005	0.029 ± 0.01	**0.026 ± 0.007**	0.028 ± 0.007	**0.026 ± 0.006**	0.057	0.038	0.343
TaW	E	**1.75 ± 0.11**	1.87 ± 0.14	2.45 ± 0.31	2.73 ± 0.53	3.21 ± 0.55	3.57 ± 0.48	2.19 ± 0.54	2.72	2.99	44.80
	F	**0.017 ± 0.002**	**0.018 ± 0.002**	0.020 ± 0.002	**0.017 ± 0.002**	**0.018 ± 0.002**	0.019 ± 0.002	**0.018 ± 0.002**	0.038	0.025	0.248
VCr	E	**1.74 ± 1.20**	2.52 ± 2.43	**2.13 ± 1.24**	**2.19 ± 0.78**	2.82 ± 1.28	3.11 ± 1.42	**1.89 ± 1.27**	**2.29**	2.82	44.80
	F	**0.016 ± 0.002**	0.018 ± 0.001	0.019 ± 0.001	**0.016 ± 0.001**	**0.017 ± 0.001**	0.018 ± 0.002	0.019 ± 0.001	0.036	0.025	0.270
VW	E	**1.32 ± 0.2**	**1.46 ± 0.16**	1.69 ± 0.21	1.90 ± 0.19	1.94 ± 0.23	2.42 ± 0.24	1.61 ± 0.16	2.50	2.00	21.30
	F	**0.014 ± 0.002**	**0.015 ± 0.002**	0.018 ± 0.003	**0.014 ± 0.002**	**0.015 ± 0.002**	0.017 ± 0.002	0.016 ± 0.002	0.037	0.023	0.292
CrW	E	**2.18 ± 0.93**	**2.45 ± 1.53**	2.76 ± 1.15	**2.31 ± 1.18**	2.84 ± 0.98	4.14 ± 1.38	3.12 ± 1.90	4.35	2.87	23.40
	F	**0.018 ± 0.004**	**0.020 ± 0.005**	0.024 ± 0.008	**0.020 ± 0.009**	**0.019 ± 0.006**	0.023 ± 0.007	0.022 ± 0.006	0.041	0.029	0.248
TaVCr	E	**0.79 ± 0.08**	0.92 ± 0.17	1.00 ± 0.24	2.26 ± 0.54	2.71 ± 0.66	3.92 ± 0.77	0.97 ± 0.13	2.43	1.97	34.10
	F	0.027 ± 0.001	0.029 ± 0.002	0.033 ± 0.002	**0.023 ± 0.002**	**0.024 ± 0.001**	0.028 ± 0.001	0.031 ± 0.002	0.054	0.045	0.313
TaVW	E	**1.00 ± 0.20**	**0.98 ± 0.18**	1.26 ± 0.23	1.80 ± 0.35	1.97 ± 0.44	2.29 ± 0.86	**0.95 ± 0.25**	1.67	1.70	39.60
	F	**0.021 ± 0.001**	0.022 ± 0.001	0.025 ± 0.001	**0.021 ± 0.002**	0.023 ± 0.001	0.026 ± 0.001	0.023 ± 0.001	0.043	0.034	0.321
TaCrW	E	**1.16 ± 0.15**	**1.28 ± 0.13**	1.58 ± 0.29	1.67 ± 0.38	1.48 ± 0.50	2.08 ± 0.57	**1.24 ± 0.11**	2.08	2.19	23.60
	F	**0.022 ± 0.001**	0.024 ± 0.001	0.027 ± 0.001	0.028 ± 0.002	0.030 ± 0.002	0.033 ± 0.002	0.026 ± 0.001	0.051	0.039	0.327
VCrW	E	**1.00 ± 0.16**	**1.07 ± 0.14**	1.37 ± 0.13	1.97 ± 0.5	2.21 ± 0.42	2.86 ± 0.64	**1.10 ± 0.14**	1.37	1.94	19.40
	F	**0.018 ± 0.001**	0.019 ± 0.001	0.022 ± 0.001	**0.017 ± 0.001**	0.019 ± 0.001	0.021 ± 0.001	0.020 ± 0.001	0.040	0.031	0.314
TaVCrW (0 K)	E	**1.22 ± 0.07**	**1.30 ± 0.10**	1.48 ± 0.16	2.26 ± 0.55	2.48 ± 0.46	3.60 ± 0.54	**1.33 ± 0.17**	2.09	2.16	50.80
	F	**0.021 ± 0.002**	**0.022 ± 0.002**	0.025 ± 0.002	**0.022 ± 0.001**	0.023 ± 0.002	0.027 ± 0.001	0.024 ± 0.002	0.049	0.037	0.488
TaVCrW (2500 K)	E	**1.63 ± 0.07**	1.74 ± 0.11	2.09 ± 0.09	2.22 ± 0.48	2.34 ± 0.59	3.68 ± 0.70	2.06 ± 0.09	2.40	2.67	59.40
	F	**0.116 ± 0.002**	0.121 ± 0.002	0.141 ± 0.003	**0.119 ± 0.007**	0.126 ± 0.006	0.150 ± 0.003	0.140 ± 0.002	0.156	0.179	0.521
Overall	E	**1.38 ± 0.09**	1.56 ± 0.21	1.80 ± 0.18	2.19 ± 0.31	2.42 ± 0.31	3.17 ± 0.28	1.67 ± 0.21	2.43	2.32	37.14
	F	**0.028 ± 0.001**	**0.029 ± 0.001**	0.034 ± 0.001	**0.029 ± 0.001**	0.030 ± 0.001	0.034 ± 0.001	0.032 ± 0.001	0.054	0.043	0.443
Deformed Structures											
TaV	E	5.47 ± 0.66	4.97 ± 0.89	5.56 ± 1.84	**3.57 ± 1.61**	**3.41 ± 1.57**	**3.63 ± 1.84**	5.80 ± 1.56	4.43	**3.63**	56.8
CrW	E	1.94 ± 0.90	2.19 ± 0.80	2.04 ± 0.84	2.74 ± 1.35	2.14 ± 1.37	2.19 ± 1.24	**1.53 ± 1.65**	1.25	**1.04**	27.1
TaCr	E	**0.81 ± 0.73**	**0.76 ± 0.66**	**0.83 ± 0.66**	2.47 ± 1.42	3.57 ± 2.11	7.10 ± 1.96	1.57 ± 1.86	1.89	**0.49**	13.3
VW	E	**0.49 ± 0.41**	**0.51 ± 0.40**	**0.65 ± 0.42**	2.90 ± 0.93	2.20 ± 0.83	3.61 ± 2.36	0.78 ± 0.38	**0.41**	0.52	66.1
TaW	E	**0.89 ± 0.56**	**1.16 ± 0.93**	**1.09 ± 0.69**	5.32 ± 1.79	8.52 ± 2.05	5.68 ± 3.68	2.82 ± 1.79	3.74	3.11	161.9
VCr	E	6.30 ± 2.65	8.49 ± 3.39	8.40 ± 1.78	7.20 ± 2.95	4.21 ± 2.17	3.50 ± 2.52	4.60 ± 3.38	4.29	**2.70**	283.2
Overall	E	2.65 ± 0.46	3.01 ± 0.48	3.10 ± 0.43	4.03 ± 0.67	4.01 ± 0.70	4.29 ± 0.73	2.85 ± 0.56	2.67	**1.91**	101.4
Inference time		51.78 ± 1.18	25.09 ± 0.02	14.59 ± 0.01	29.48 ± 0.23	15.37 ± 0.04	4.43 ± 0.00	14.97 ± 0.09	17.57	7.25	0.50
Memory consumption		36.78 ± 0.00	16.93 ± 0.00	8.48 ± 0.00	28.82 ± 0.00	13.87 ± 0.00	5.91 ± 0.00	13.15 ± 0.00	–	–	–

[a] Inference times for MTP and EAM were measured on two Intel Xeon E6252 Gold (Cascade Lake) CPUs. For GM-NN, an NVIDIA GeForce RTX 3090 Ti 12GB GPU was used, while all other models were evaluated using an NVIDIA A100 GPU with 80 GB of RAM.

https://doi.org/10.52202/079017-3940

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction include the presented work's main contributions, assumptions, and results. Each of these claims are backed up through theoretical and empirical results in the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of the presented work in Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We provide the proof of the equivariance and traceless property of MPNNs based on irreducible Cartesian tensors in Appendices C and D.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We describe the entire experimental setting and all details on the performed experimens in Section 5 and Appendix E. We also provide a detailed description of MPNNs employed in this work, including the weight initialization; see Section 4.2 and Appendix B.2. Furthermore, we describe the construction of irreducible Cartesian tensor products and introduce the irreducible Cartesian tensor products in Section 4.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

 https://doi.org/10.52202/079017-3940

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The source code is available on GitHub and can be accessed via this link: `https://github.com/nec-research/ictp`. All data sets used in this study are publicly available: rMD17 (`https://doi.org/10.6084/m9.figshare.12672038.v3`), MD22 (`http://www.sgdml.org`), 3BPA (`https://github.com/davkovacs/BOTNet-datasets`), acetylacetone (`https://github.com/davkovacs/BOTNet-datasets`), and Ta–V–Cr–W (`https://doi.org/10.18419/darus-3516`).

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The details on the experimental setting are provided in Section 5 and Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We run all experiments using several random seeds and provide the corresponding standard deviations and error bars in all tables and figures.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide details on the computer resources in Section 5 and Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The presented work follows the NeurIPS Code of Ethics. We also include Appendix F, which discusses the broader social impact.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

https://doi.org/10.52202/079017-3940

Answer: [Yes]

Justification: We include Appendix F, which discusses the broader social impact.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper does not release any data or models with a high risk for misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We reference all employed data sets and source codes in the main text.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide a detailed description of the construction of irreducible Cartesian tensors, computation of their irreducible tensor products, and the resulting equivariant message-passing layers; see Section 4 and Section B. We also provide training details in Appendix E.2 and describe the employed data sets in greater detail in Appendix E.1. The source code includes comprehensive documentation as well.

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We performed neither crowdsourcing nor research with human subjects.

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We performed neither crowdsourcing nor research with human subjects.

    Guidelines:

https://doi.org/10.52202/079017-3940

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.