nature communications

Article | Article in Press

https://doi.org/10.1038/s41467-026-77263-z

Atomistic machine learning with irreducible Cartesian natural tensors

Received: 4 May 2026

Accepted: 19 August 2026

Published online: 27 August 2026

Check for updates

Qun Chen[1], A. S. L. Subrahmanyam Pattamatta[2], Boyu Wang[1], David J. Srolovitz[2] & Mingjian Wen[1]

Atomistic machine learning is a powerful tool for accurate and efficient investigation of material behavior at the atomic scale. While attempts have been made to construct models directly within Cartesian space, they face challenges in providing a systematic framework based on irreducible representations—a core feature of widely used spherical models. Here we propose Cartesian Natural Tensor Networks to overcome these limitations and thus offer a general, symmetry-preserving framework for atomistic machine learning. We present a theory of irreducible representations using Cartesian natural tensors, comprising their construction, their products, and a systematic scheme to decompose and reconstruct high-rank physical tensors. Leveraging this machinery, we develop equivariant machine learning interatomic potentials for materials and molecular systems with performance on par with leading spherical models. It further captures accurate structure–property relationships for tensorial quantities ranging from low-rank dipole moments to high-rank tensors with complex symmetries, such as the elastic constant tensor.

Atomistic machine learning (ML) represents a data-driven paradigm that learns to predict material and molecular properties directly from the atomic structure. The primary inputs are the spatial coordinates of atoms and their chemical identities, while the outputs span interatomic potential energies and forces[1,2]; scalar properties such as bond dissociation energies and band gaps[3,4]; and tensorial quantities including dipole moments, polarizabilities, and elastic constant tensors[5,6]. When trained on quantum-mechanical data, these models approach ab initio accuracy at a fraction of the computational cost, enabling simulations of larger systems and longer time scales, and systematic exploration across vast chemical and materials spaces. By uniting accurate physical data with scalable learning approaches, atomistic ML underpins transferable interatomic potentials, rapid property screening, and the construction of robust structure–property relationships. Atomistic ML has become foundational in the sciences and engineering, accelerating rational discovery and design across diverse fields, including energy conversion and storage[7,8], heterogeneous catalysis[9,10], and pharmaceutical discovery[11,12].

At the core of any atomistic ML model lies the central task of describing the local atomic environment around each atom. This step is critical to any atomistic ML because it converts raw atomic coordinates and chemical identities into numerical representations that ML algorithms can process. Early methods relied on descriptors designed to capture important structural features like bond lengths and bond angles (e.g., atom-centered symmetry functions[13], the Coulomb matrix[14], and DeePMD descriptors[15]). More recently, the field has evolved toward more systematic approaches based on mathematical expansions of the atomic environment using well-established basis functions. Current state-of-the-art approaches use two key mathematical tools: polynomials and/or trigonometric functions to describe distance relationships between atoms, and spherical harmonics to capture angular relationships. Examples include SNAP[16], ACE[17], TFN[18], NequIP[19], MACE[20], and GRACE[21]. In other words, these methods first transform atomic structures from their Cartesian coordinates (i.e., {x, y, z} positions) into spherical coordinates (distances and angles), and then train the ML model in this spherical representation. However,

[1]Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu, China. [2]Department of Mechanical Engineering, The University of Hong Kong, Hong Kong SAR, China. e-mail: mjwen@uestc.edu.cn

since atomic structures are naturally described in Cartesian coordinates for most simulation methods and property calculations, there is a compelling motivation to develop methodologies that work directly in Cartesian space, directly utilizing geometric information in its natural form while maintaining clear physical interpretability.

Considerable progress has been made in developing atomistic ML methods that operate directly in Cartesian space. For example, models such as MTP[22], REANN[23], CACE[24], CAMP[25], and ICTP[26] build on the ideas of Cartesian atomic moments or Cartesian cluster expansion to construct interatomic potentials. FieldSchNet[27], TensorNet[28], HotPP[29], and TACE[30] utilize Cartesian representations to predict low-rank tensorial properties. Despite these achievements, Cartesian approaches still face key challenges that limit their broader applicability. First, models employing reducible Cartesian tensors, such as HotPP[29] and CAMP[25], naturally accommodate arbitrary-rank tensors and respect global rotational symmetry, but do not systematically enforce the internal index symmetries of physical tensors (e.g., $C_{ijkl} = C_{jikl} = C_{klij}$ of the elastic constant tensor), leaving unphysical redundancies in the predicted tensors. Second, models employing irreducible Cartesian representations, such as TensorNet[28], ICTP[26], and TACE[30], could in principle be extended to higher ranks, but in practice remain limited to low-rank tensorial properties due to the absence of a systematic scheme for the decomposition and reconstruction of physical tensors using irreducible Cartesian building blocks. These limitations motivate a principled and systematic framework to fully harness the advantages of working directly in Cartesian space.

This work addresses these challenges by developing a comprehensive computational framework for atomistic ML using Cartesian natural tensors. A natural tensor is one whose components are determined by the geometric properties of the space in which it lives and is, therefore, an intrinsic geometric construction rather than a regular tensor. Natural tensors find wide application in fluid physics (e.g., Navier–Stokes equations)[31], electromagnetism (e.g., multipole expansions of charge distributions)[32], and increasingly in machine learning (e.g., equivariant neural networks)[26,28]. In this work, we adapt the theory of irreducible Cartesian natural tensors for atomistic ML, further develop a systematic scheme for the decomposition and reconstruction of arbitrary physical tensors, and provide the mathematical tools and software needed to apply this framework to atomic systems.

Building on this foundation, we introduce a modeling framework applicable to both interatomic potentials and structure–property relationships of high-rank tensorial observables with intrinsic symmetries. Our Cartesian Natural Tensor Networks (CarNet) framework is E(3)-equivariant; i.e., it respects the three-dimensional rotational, translational, and inversion symmetries inherent to atomic systems. We demonstrate the capabilities of CarNet through a range of atomistic ML tasks. As an MLIP, CarNet achieves accuracy comparable to leading spherical-tensor approaches across both materials and molecular systems, with universal models that generalize across the periodic table. Beyond potentials, CarNet provides a unified route to predicting tensorial properties of varying rank and symmetry, including dipole moments, polarizabilities, nuclear magnetic shielding, and elastic constant tensors. CarNet enables the exploration of the rich landscape of Cartesian representations and their applications in atomistic ML.

Results

Cartesian natural tensor theory

A natural tensor is a fully symmetric Cartesian tensor whose traces vanish on any pair of indices[33–35]. Formally, a rank-n tensor $\mathbf{X}_n$ with components $X_{i_1 i_2 \ldots i_n}$ is natural if it is

1. Symmetric: $X_{i_{\pi(1)} i_{\pi(2)} \cdots i_{\pi(n)}} = X_{i_1 i_2 \ldots i_n}$ for all $\pi \in S_n$, where S_n is the symmetric group of degree n (the group of all permutations of n indices),
2. Traceless: $\delta_{i_a i_b} X_{i_1 \ldots i_a \ldots i_b \ldots i_n} = 0$ for all $1 \le a < b \le n$,where δ_{ij} is the Kronecker delta and Einstein summation is implied over repeated indices (the same below unless otherwise stated). Scalars (rank-0 tensors) and vectors (rank-1 tensors) are trivially natural tensors.

Natural tensors are widely used to represent physical properties. For example, any rank-2 Cartesian tensor $\mathbf{T}$ (T_{ij}) can be decomposed into

$$T_{ij} = \underbrace{\frac{1}{3} T_{kk} \delta_{ij}}_{n=0} + \underbrace{\frac{1}{2}\left(T_{ij} - T_{ji}\right)}_{n=1} + \underbrace{\frac{1}{2}\left(T_{ij} + T_{ji}\right) - \frac{1}{3} T_{kk} \delta_{ij}}_{n=2}, \tag{1}$$

where the $n = 0, 1, 2$ natural tensors are typically called the isotropic, antisymmetric, and symmetric parts, respectively (Fig. 1c). This decomposition is useful because each part has a distinct physical meaning. For example, if $\mathbf{T}$ is a stress tensor, the isotropic part represents the hydrostatic pressure (causing volumetric change), the antisymmetric part is related to torques (zero in classical continuum mechanics), and the symmetric part represents the shear stress (causing shape change). Conversely, given three natural tensors of ranks 0, 1, and 2, one can fully reconstruct a rank-2 Cartesian tensor.

In three dimensions, a rank-n natural tensor $\mathbf{X}_n$ has 3^n components, but only $2n + 1$ of these are independent (due to the symmetry and traceless constraints). A rank-n natural tensor $\mathbf{X}_n$ furnishes a $2n + 1$-dimensional irreducible representation of the special orthogonal group SO(3) (the group of 3D rotations). Intuitively, under rotations, the $2n + 1$ independent components mix in a way that prevents decomposition into smaller sets of components that transform independently under SO(3); formally, the representation space of $\mathbf{X}_n$ is irreducible under SO(3), meaning it contains no proper nontrivial SO(3)-invariant subspaces[36]. This makes natural tensors particularly well-suited for atomistic ML, where rotational equivariance is crucial. Below, we summarize the three fundamental natural tensor operations underlying our framework: constructing natural tensors from a unit vector, computing products of natural tensors, and decomposing and reconstructing physical tensors. Further details are provided in Supplementary Note 1.

Given a unit vector $\hat{\mathbf{r}}$, a rank-n natural tensor $\mathbf{X}_n$ can be constructed in two steps (Fig. 1a). 1. Symmetric polyadic tensor: Form the rank-n tensor $\mathbf{U} = \mathbf{r}^{\otimes n} = \hat{\mathbf{r}} \otimes \hat{\mathbf{r}} \otimes \cdots \otimes \hat{\mathbf{r}}$ (repeated n times). By construction, $\mathbf{U}$ is fully symmetric. 2. Trace removal: Project $\mathbf{U}$ into the symmetric traceless subspace via $\mathbf{X}_n = \mathbf{H} \odot^n \mathbf{U}$, where $\mathbf{H}$ is the rank-$2n$ projection tensor, and $\odot^n$ denotes n-fold contraction over n pairs of indices (i.e., $X_{k_1 \ldots k_n} = H_{k_1 \ldots k_n i_1 \ldots i_n} U_{i_1 \ldots i_n}$). $\mathbf{H}$ is given by[37]

$$H_{k_1 \ldots k_n i_1 \ldots i_n} = \sum_{t=0}^{\lfloor n/2 \rfloor} (-1)^t \cdot F \cdot \{\delta_{ki}^{\otimes n-2t} \delta_{kk}^{\otimes t}\} \delta_{ii}^{\otimes t}, \tag{2}$$

where $F = \frac{(2n-2t-1)!!}{(2n-1)!!}$, the superscript $^{\otimes t}$ is a shorthand notation for the t-fold tensor product of Kronecker deltas (e.g., $\delta_{ki}^{\otimes 2} = \delta_{k_1 i_1} \delta_{k_2 i_2}$ and $\delta_{ii}^{\otimes 2} = \delta_{i_1 i_2} \delta_{i_3 i_4}$), and the curly braces $\{\}$ denote symmetrization achieved by summing over all unique permutations of the indices. For example, when $n = 2$, we have $H_{k_1 k_2 i_1 i_2} = \delta_{k_1 i_1} \delta_{k_2 i_2} - \frac{1}{3} \delta_{k_1 k_2} \delta_{i_1 i_2}$, and the corresponding rank-2 natural tensor is $X_{k_1 k_2} = H_{k_1 k_2 i_1 i_2} U_{i_1 i_2} = r_{k_1} r_{k_2} - \frac{1}{3} \delta_{k_1 k_2} r_{i_1} r_{i_1}$. The projection tensor $\mathbf{H}$ is analogous to spherical harmonics that generate irreducible representations of SO(3) from a unit vector, but it is expressed entirely in Cartesian form.

The tensor product between two natural tensors can be expressed as a direct sum of a set of natural tensors[35,38]. Given natural tensors $\mathbf{X}_{l_1}$ of rank l_1 and $\mathbf{Y}_{l_2}$ of rank l_2, their product

$$\mathbf{Z}_{l_3} = \mathbf{X}_{l_1} \hat{\otimes} \mathbf{Y}_{l_2} \tag{3}$$

is a tensor whose rank lies in the range $|l_1 - l_2| \le l_3 \le l_1 + l_2$ (analogous to spherical tensors[39]), where $\hat{\otimes}$ represents the natural tensor product

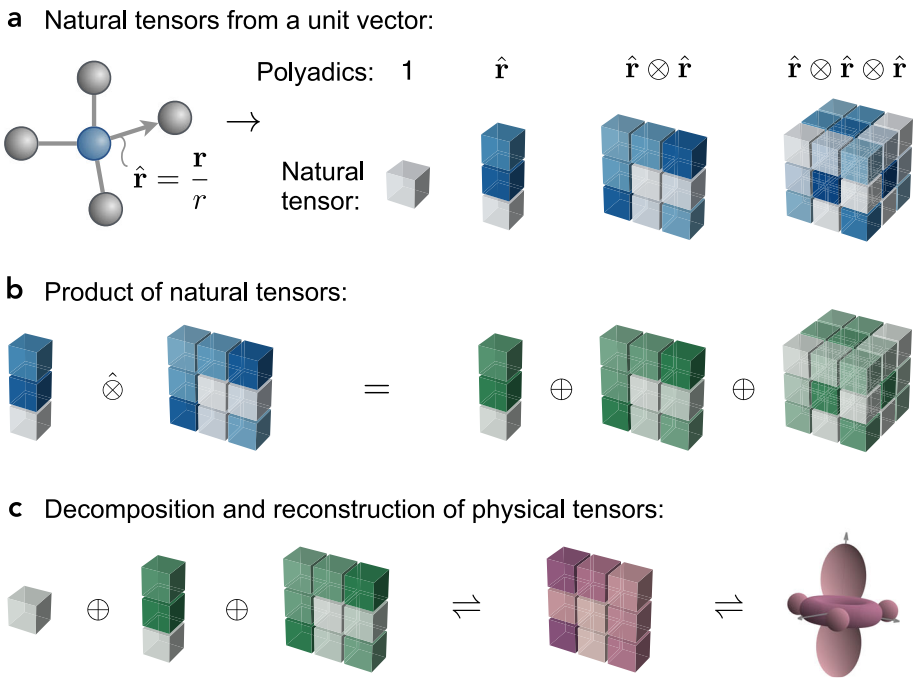

Fig. 1 | Schematic illustration of Cartesian natural tensor operations. **a** Construction of natural tensors of different ranks from a unit vector $\hat{\mathbf{r}}$. **b** Tensor product between a rank-1 and a rank-2 natural tensor generates three natural tensors of ranks 1, 2, and 3. **c** Any physical tensor (e.g., the nuclear shielding tensor) can be decomposed into a set of natural tensors and, conversely, reconstructed from them. ⊗ denotes the product between ordinary tensors, $\hat{\otimes}$ denotes the product between natural tensors, and ⊕ denotes the direct sum of natural tensors.

(e.g., when $l_1 = 1$ and $l_2 = 2$, $\mathbf{Z}_{l_3}$ is of rank 1, 2, or 3; see Fig. 1b). To obtain $\mathbf{Z}_{l_3}$, we can first compute the ordinary tensor product $\mathbf{W} = \mathbf{X}_{l_1} \otimes \mathbf{Y}_{l_2}$, and then symmetrize $\mathbf{W}$ and remove the traces. Similar to the construction of natural tensors from unit vectors, based on the formulas in ref. 38, we have derived an explicit expression for a rank-$(l_1 + l_2 + l_3)$ projection tensor $\mathbf{H}$ that performs this operation. For even $l_1 + l_2 - l_3 = 2d$, we have

$$H_{k_1\ldots k_{l_3} i_1\ldots i_{l_1} j_1\ldots j_{l_2}} = \sum_{t=0}^{\min(l_1,l_2)-d} (-2)^t \cdot F \cdot \{\delta_{ik}^{\otimes l_1-(d+t)} \delta_{jk}^{\otimes l_2-(d+t)} \delta_{kk}^{\otimes t}\} \delta_{ij}^{\otimes d+t}, \quad (4)$$

where $F = \frac{(2l_3-2t-1)!!}{(2l_3-1)!!}$. For odd $l_1 + l_2 - l_3 = 2d + 1$, the formula is similar but with the multiplication of an additional Levi–Civita symbol (expression given in Supplementary Note 1). With this, $\mathbf{Z}_{l_3}$ can be computed as $\mathbf{Z}_{l_3} = \mathbf{H} \odot^{l_1} \mathbf{X}_{l_1} \odot^{l_2} \mathbf{Y}_{l_2}$ (i.e., $Z_{k_1\ldots k_{l_3}} = H_{k_1\ldots k_{l_3} i_1\ldots i_{l_1} j_1\ldots j_{l_2}} X_{i_1\ldots i_{l_1}} Y_{j_1\ldots j_{l_2}}$). Here, the projection tensor $\mathbf{H}$ is analogous to the Clebsch–Gordan coefficients[39] for spherical tensor product.

A central task in utilizing natural tensors involves converting between regular Cartesian tensors and their natural representations. The decomposition and reconstruction in Eq. (1) for rank-2 tensors are possible for arbitrary tensors; however, the complexity increases significantly with the rank and internal index symmetries of the tensor. Unlike the rank-2 case in Eq. (1), higher-rank tensors yield multiple linearly dependent decomposition candidates[34], which should be carefully selected to form a linearly independent basis set of natural tensors. Moreover, the internal index symmetries of a physical tensor impose additional constraints on the form of natural tensors[37]. For example, a generic rank-4 tensor yields six rank-3 natural tensor candidates in its decomposition spectrum, but only three of them are linearly independent. The rank-4 elastic constant tensor $\mathbf{C}$ has symmetries $C_{ijkl} = C_{jikl} = C_{klij}$, under which all associated rank-3 candidates vanish. Several studies have examined the decomposition of physical tensors with specific ranks[37,40–42]; a more general approach was proposed by Coope et al.[33–35]. However, a general method for tensors of arbitrary rank and symmetry remains unavailable (to our knowledge). We have developed a systematic approach to addressing the challenges, which consists of three key steps.

First, create the rank-$(m + n)$ tensor $\mathbf{G}$ to map between the rank-n physical tensor space and the rank-m natural tensor space[34]:

$$G_{k_1\ldots k_m i_1\ldots i_n} = \sum_{t=0}^{\lfloor m/2 \rfloor} c_t \{\{\delta_{ki}^{\otimes(m-2t)} \delta_{kk}^{\otimes t} \delta_{ii}^{\otimes t}\}\} \delta_{ii}^{\otimes(n-m)/2}, \quad (5)$$

where c_t are known coefficients (given in Supplementary Note 1), and the double curly braces $\{\{\}\}$ denote symmetrization achieved by averaging over all unique permutations of the indices. Depending on the assignment of the indices, there exist multiple candidate mapping tensors for a given n and m. For example, when $n = 3$ and $m = 1$, we have $G_{k_1 i_1 i_2 i_3} = c_0 \delta_{ki} \delta_{ii}$ (no Einstein summation on i), resulting in three candidates: $G^1 = c_0 \delta_{k_1 i_1} \delta_{i_2 i_3}$, $G^2 = c_0 \delta_{k_1 i_2} \delta_{i_3 i_1}$, and $G^3 = c_0 \delta_{k_1 i_3} \delta_{i_1 i_2}$, where the subscript $k_1 i_1 i_2 i_3$ is dropped for clarity.

Next, identify a subset of N_g unique linearly independent mapping tensors from all N candidates $\mathbf{G}^1, \ldots, \mathbf{G}^N$. This can be achieved via a QR factorization[43], followed by symmetry-informed elimination according to the intrinsic symmetries of the physical tensor. For example, when $n = 3$ and $m = 1$, a QR factorization will verify that all three candidates above are linearly independent. But for a rank-3 tensor with symmetry $T_{i_1 i_2 i_3} = T_{i_1 i_3 i_2}$, $\mathbf{G}^2$ and $\mathbf{G}^3$ yield identical outcomes upon contraction with $\mathbf{T}$; thus, only $\mathbf{G}^1$ and $\mathbf{G}^2$ are retained as unique mapping tensors.

Finally, construct the decomposition projector $\mathbf{H}$ and embedding tensor $\mathbf{Q}$ to map between a physical tensor with arbitrary rank and

symmetry and its natural representation:

$$\begin{aligned}\mathbf{H}^{p}_{m+n} &= \sum_{q=1}^{N} h_{pq}\mathbf{G}^{q}_{m+n}\\ \mathbf{Q}^{p}_{m+n} &= \mathbf{G}^{p}_{m+n} + \sum_{q=N_g+1}^{N} \beta_{qp}\mathbf{G}^{q}_{m+n},\end{aligned} \tag{6}$$

where $p = 1, 2, \ldots, N_g$, and h_{pq} and β_{qp} (expressions given in Supplementary Note 1) are coefficients that can be determined by enforcing the orthonormal condition between different $\mathbf{G}^q$. With them, one can utilize $\mathbf{X}^{p}_{m} = \mathbf{H}^{p}_{m+n} \odot^{n} \mathbf{T}_n$ to extract the natural tensors from an ordinary tensor, and, conversely, utilize

$$\mathbf{T}_n = \sum_{m=0}^{n}\sum_{p=1}^{N_g} \mathbf{Q}^{p}_{m+n} \odot^{m} \mathbf{X}^{p}_{m} \tag{7}$$

to reconstruct an ordinary tensor from its natural representations.

In summary, our theoretical contributions are twofold. First, we have developed a systematic approach to decomposing and reconstructing physical tensors of arbitrary rank and symmetry using natural tensors, which is not available in the literature (to our knowledge). Second, we have derived explicit expressions for the projection tensors (Eqs. (2), (4), and (6)) that consist of only Kronecker deltas and Levi–Civita symbols, which can be precomputed and stored, enabling efficient natural tensor operations.

The CarNet model

Leveraging the three fundamental operations of natural tensors and the concept of moment tensors as introduced in MTP[22] and CAMP[25], we propose CarNet: a theoretically grounded framework for constructing equivariant atomistic ML models. CarNet takes an atomic structure as input, generates equivariant atom features, and produces interatomic potentials or structure–property relationships, including those involving high-rank tensors. A schematic overview of the model architecture is illustrated in Fig. 2.

The model begins by encoding the atomic structure. The relative position vector $\mathbf{r}$ between an atom and its neighbor is decomposed into its scalar distance r and the unit directional vector $\hat{\mathbf{r}} = \mathbf{r}/r$. The scalar distance r is then expanded into R using a set of radial basis functions, specifically Chebyshev polynomials of the first kind. Angular information is incorporated through $\hat{\mathbf{r}}$, which is transformed into natural tensors $\mathbf{X}$ via the first operation described in Section 2.1. Atomic numbers z are embedded into learnable initial atom features $\mathbf{h}$, completing the initial encoding of the atomic structure.

The core architecture of CarNet comprises a multilayer graph neural network (GNN) that iteratively refines atom features. Within each GNN layer, the atomic moment $\mathbf{M}$ is constructed as a tensor product involving radial function R, current atom features $\mathbf{h}$, and the angular component $\mathbf{X}$. Subsequently, a self-tensor product is performed on $\mathbf{M}$, yielding the hyper moment $\mathbf{H}$ that encodes many-body interactions[17,20,22]. Both steps employ the natural tensor product formalism introduced above, ensuring that $\mathbf{M}$ and $\mathbf{H}$ are natural tensors. Empirical results (below) indicate that typically two to three GNN layers suffice to attain high predictive accuracy.

At the final stage, relevant natural tensors derived from atom features $\mathbf{h}$ are selected to construct the desired physical quantities. For interatomic potentials, this process involves extracting the rank-0 natural tensor (scalar). For higher-rank tensorial properties, the relevant natural tensors are extracted from atom features according to their decomposition and reconstruction spectrum, and the target physical tensor is reconstructed from these components using the third operation in Section 2.1.

The overall model architecture, the reconstruction of atomic and structural physical tensors, and the training procedures are provided in the Methods section. A detailed comparison between CarNet and existing Cartesian methods for atomistic ML (e.g., TensorNet[28], HotPP[29], ICTP[26], and TACE[30]) is given in Supplementary Note 2.

Bulk LiPS and water

We first apply CarNet to develop interatomic potentials for periodic systems—specifically, inorganic lithium phosphorus sulfide (LiPS) (a solid-state electrolyte)[19] and bulk liquid water[44]. We assess the model's performance by computing the mean absolute error (MAE) or root mean square error (RMSE) of energies and atomic forces on the test sets, benchmarking against state-of-the-art models in the literature, including both spherical models (e.g., NequIP[19] and MACE[20]) and Cartesian models (e.g., CACE[24] and CAMP[25]). For both systems, CarNet configured with two GNN layers achieves the lowest reported errors, as shown in Table 1; increasing to three layers further reduces errors.

Beyond low MAEs, CarNet enables stable molecular simulations to accurately predict structural and dynamical properties. While low errors in energy and forces are necessary, they are not sufficient to guarantee the physical reliability of molecular dynamics (MD)

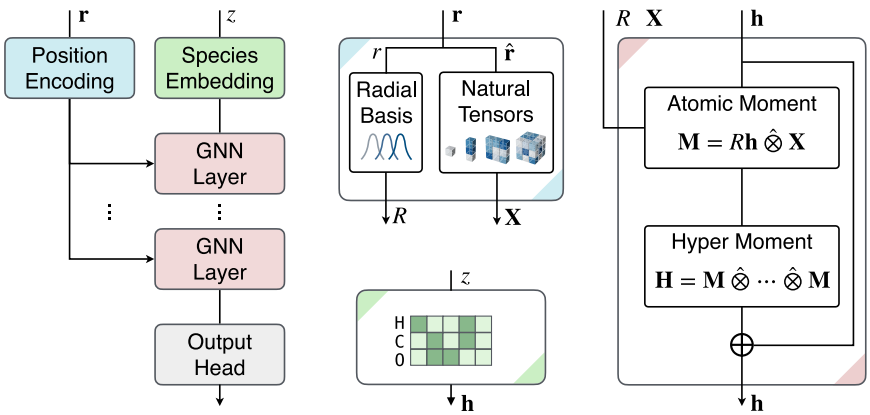

Fig. 2 | Overview of the CarNet model architecture. The relative distance vector $\mathbf{r}$ of an atom from its neighbor is encoded using a set of radial basis functions for its magnitude r and natural tensors constructed from polyadics of the unit vector $\hat{\mathbf{r}}$. The atomic species z is encoded using a learnable embedding to generate the initial atom features $\mathbf{h}$. With the radial part R, the angular part $\mathbf{X}$, and the atom features $\mathbf{h}$, each graph neural network (GNN) layer first constructs the atomic moment $\mathbf{M}$ and then the hyper moment $\mathbf{H}$ using natural tensor products (denoted by $\hat{\otimes}$). Finally, the atomic features are mapped to the target properties using an output head.

Table 1 | Model performance on the LiPS and water datasets

Dataset	Model	Energy (meV)	Forces (meV/Å)
LiPS	NequIP[19]	0.12	7.7
	CAMP[25]	0.12	7.4
	CarNet (2 layers)	0.11	6.4
	CarNet (3 layers)	**0.09**	**5.6**
Water	BPNN[13]	2.3	120
	ACE[17]	1.7	99
	REANN[23]	0.8	53
	DeePMD[15]	2.1	92
	NequIP[19]	0.94	45
	MACE[20]	0.63	36
	CACE[24]	0.59	47
	CAMP[25]	0.59	34
	CarNet (2 layers)	**0.54**	34
	CarNet (3 layers)	**0.54**	**31**

For LiPS, mean absolute errors (MAEs) of energy and forces on the test set are reported. For water, root-mean-square errors (RMSEs) of energy and forces are reported. CAMP results are from ref. 25; CACE and others are from ref. 24.
Bold text indicates the smallest error.

Table 2 | Prediction errors in ethanol properties

	$\mathcal{E}$ (kcal mol^{-1})	F (kcal mol^{-1} Å^{-1})	μ (D)	α (Bohr^3)	δ_{all} (ppm)	σ_{all} (ppm)
PaiNN[80]	0.027	0.150	0.003	0.009	–	–
FieldSchNet[27]	0.017	0.128	0.004	0.008	0.169	–
TensorNet[28]	0.008	0.058	0.003	0.007	0.139	–
CarNet (multitask)	0.0063	0.034	0.0011	**0.0051**	0.044	0.065
CarNet (L = 2)	0.0054	0.027	0.0009	0.0076	0.037	0.057
CarNet (L = 3)	**0.0048**	**0.022**	**0.0007**	0.0063	**0.031**	**0.047**

Mean absolute errors (MAEs) on the test set are reported for energy $\mathcal{E}$, forces **F**, dipole moment $\boldsymbol{\mu}$, polarizability $\boldsymbol{\alpha}$, nuclear chemical shift δ_{all} and shielding tensor $\boldsymbol{\sigma}_{all}$ for all atoms. L is the maximum rank of the natural tensors employed.
Bold text indicates the smallest error.

simulations; unphysical force predictions can lead to instability or drift during simulations despite good MAEs[45].

To evaluate the stability and physical fidelity, we first perform MD simulations for LiPS under an NVT ensemble. For LiPS, five MD simulations with different initial velocities were carried out using a timestep of 1 fs over a total duration of 300 ps (simulation details in Methods). No instabilities or trajectory collapse were observed in any of the MD runs. We next analyze the radial distribution functions (RDF) derived from the MD trajectories. The RDF (Fig. 3b) predicted by CarNet closely matches AIMD simulation results[44]. We also compute the diffusion coefficient from the mean squared displacement (MSD) of the MD trajectories. The MSD of Li^+ in LiPS (Fig. 3c) exhibits linear behavior, indicative of normal diffusion. The calculated diffusion coefficient from the five MD runs is $D = (1.18 \pm 0.19) \times 10^{-5}$ cm^2/s, consistent with AIMD estimates of $D = 1.37 \times 10^{-5}$ cm^2/s[19].

We also conducted MD simulations for water using setups similar to those for LiPS (simulation details in Methods). The RDF of oxygen-oxygen pairs in water at 300 K (Fig. 3e) agrees well with X-ray[46] and neutron[47] diffraction data. The predicted diffusion coefficient is $D = 2.68 \times 10^{-5}$ cm^2/s at 300 K as compared with AIMD results of $D = 2.67 \times 10^{-5}$ cm^2/s[44]. To further check the stability of CarNet, following[24], we performed MD simulations at high temperatures up to 2000 K. At high temperatures, the MD simulations remained stable, and the MSD exhibited linear behavior (Fig. 3f). This demonstrates CarNet's superior stability. We do not expect physical quantities (e.g., D in Fig. 3f and RDF in Supplementary Fig. 1) at high temperatures to be quantitatively accurate; this example serves only to demonstrate the stability of CarNet.

Molecular ethanol

We also evaluated the performance of CarNet for ethanol as a small-molecule test. In addition to energies and atomic forces, the ethanol dataset[27] includes three tensorial properties: dipole moment $\boldsymbol{\mu}$, polarizability $\boldsymbol{\alpha}$, and nuclear shielding tensor $\boldsymbol{\sigma}$. The dipole moment $\boldsymbol{\mu}$ is a rank-1 structural tensor defined at the molecular level, representing a property of the entire molecule. The polarizability $\boldsymbol{\alpha}$ is a rank-2 structural tensor, also defined at the molecular level. In contrast, the nuclear shielding $\boldsymbol{\sigma}$ is a rank-2 atomic tensor; i.e., each atom in the molecule has a shielding tensor that is sensitive to the local electronic environment. The nuclear chemical shift is a scalar property derived from $\boldsymbol{\sigma}$ as $\delta = \mathrm{Tr}(\boldsymbol{\sigma})/3$.

CarNet demonstrates significant improvements in predicting these properties relative to existing state-of-the-art models such as FieldSchNet[27] and TensorNet[28]. Using a multitask learning framework with a shared backbone and multiple output heads to concurrently learn all properties, CarNet achieves the highest accuracy across all tasks (Table 2). For instance, the errors in predicting the dipole moment $\boldsymbol{\mu}$ and nuclear chemical shift δ_{all} are ~3× smaller than those from TensorNet. Separate errors δ_H, δ_C, and δ_O for each element are given in Supplementary Note 3. This highlights the model's ability to learn a unified representation for predicting distinct scalars, atomic tensors, and structural tensors. We also trained models where each property was learned individually (energy and forces jointly), observing reductions in errors across all outputs except $\boldsymbol{\alpha}$. Models employing natural tensor representations with a maximum rank of $L = 3$ outperform their $L = 2$ counterparts, underscoring the critical role of high-rank tensors in capturing complex interatomic interactions, while highlighting limitations of models that rely on tensor representations up to rank-1 and rank-2 (e.g., FieldSchNet and TensorNet).

Crystal elastic constant tensor

We now demonstrate the capability of CarNet to predict the elastic constant tensor (characterizing linear elastic response of a material under applied stress) of inorganic crystalline solids. Currently, no Cartesian atomistic ML methods can model the full (rank-4) elastic constant tensor (with up to 21 independent components depending on crystal symmetry). This is more challenging due to both the inherent complexity of the elastic constant tensor and the compositional diversity of the dataset. The dataset[6] encompasses structures spanning all seven crystal systems, involving 84 chemical elements, thereby presenting substantial variability in symmetry, composition, and structural complexity. CarNet effectively handles this complexity, exhibiting robust applicability across diverse crystal symmetries and compositional variations.

CarNet directly predicts the full elastic constant tensor while inherently satisfying the two fundamental physical constraints: (1) frame indifference (coordinate system invariance) and (2) crystal symmetry adherence. Frame indifference ensures the predicted tensor is equivariant under rigid rotations of the coordinate system, while the symmetry constraint guarantees that the tensor reflects the intrinsic point-group symmetries of the crystal[6]. From the predicted rank-4 elastic constant tensor, we can obtain the Voigt stiffness matrix C[48]. Scalar elastic moduli such as the bulk modulus K, shear modulus G, and Young's modulus E are derived from C using the Hill averaging scheme[49]. Quantitatively, CarNet achieves MAEs of 5.31 for K, 6.39 for G, and 13.68 for E over the 84 elements in the dataset (where values range from near 0 to 400 GPa for K and G and up to ~800 GPa for E; see Fig. 4a). The errors are ~18% lower than those by MatTen[6] and are

https://doi.org/10.1038/s41467-026-77263-z

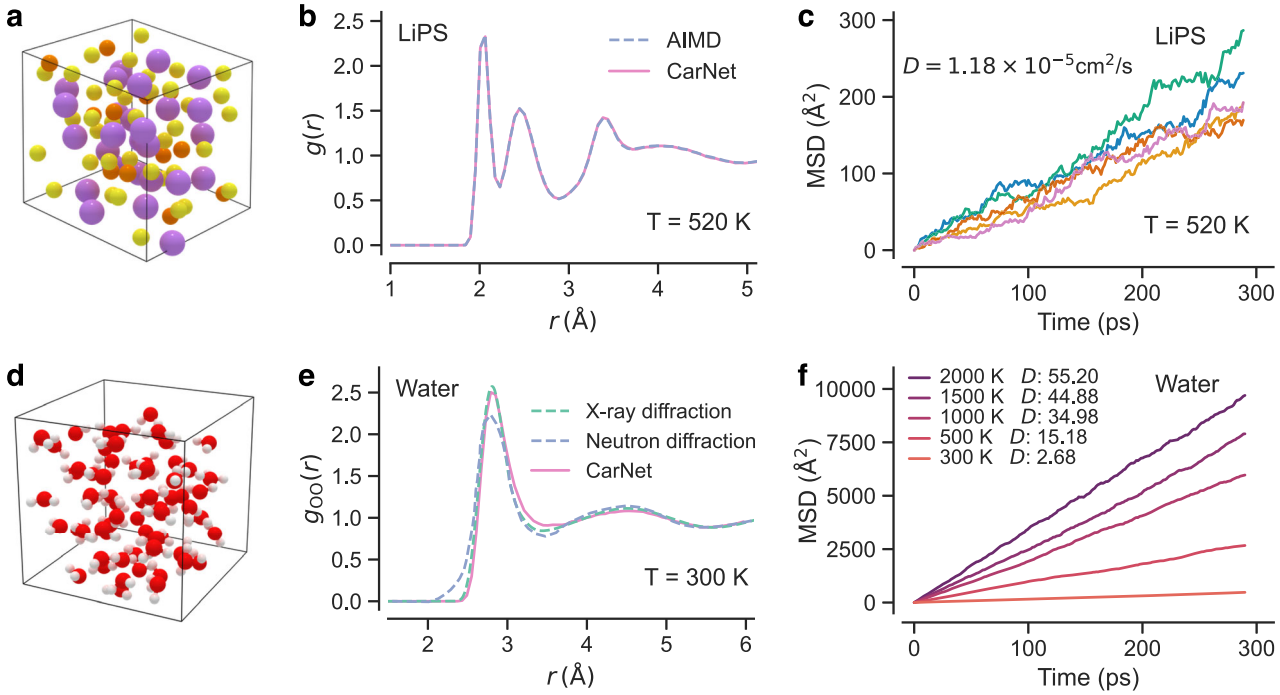

Fig. 3 | Molecular dynamics (MD) simulation results of bulk LiPS and water systems. a–c Crystal structure, radial distribution function (RDF), and mean squared displacement (MSD) of Li^+ ions versus time for LiPS. **d–f** Simulation cell, RDF of oxygen-oxygen pairs, and MSD versus time for bulk water. The reference ab initio molecular dynamics (AIMD) and experimental results are at the same temperatures shown in the figures, except for the X-ray diffraction data, which is at 295 K. In panel (**c**), five MD simulations with different initial velocities were performed, and the average diffusion coefficient D is reported. The MD simulation for water used a 2 × 2 × 2 replication of the unit cell shown in panel (**d**). In panel (**f**), D is given in the units of $\times 10^{-5}$ cm^2/s. Simulation cells are plotted using AtomViz[82]; atom colors: purple (Li), orange (P), yellow (S), red (O), and white (H). Source data are provided as a Source Data file.

comparable to those by XPaiNN[50], both of which are spherical models specifically designed for elastic constant tensors (Table 3).

A detailed analysis reveals that CarNet maintains consistent predictive accuracy across different crystal systems. The relative error, MAE normalized by the mean absolute deviation (MAD) of the reference values, is comparable among all seven crystal systems (Fig. 4b). Despite dataset imbalance (see Supplementary Fig. 2) favoring high-symmetry over low-symmetry crystals (e.g., cubic vs. triclinic), CarNet demonstrates effective transferability and generalization, indicating its ability to learn unified representations across a broad structural spectrum.

Beyond scalar moduli K, G, and E, the ability to predict the full elastic constant tensor enables efficient analysis of anisotropic elastic behaviors. For example, the directional Young's modulus E_d can be computed for all orientations, enabling comprehensive evaluation of elastic anisotropy and directional stiffness variations (Fig. 4c). This capability underscores the utility of CarNet for materials design and mechanical property optimization where directional elastic responses are critical.

Universal MLIP for materials

To further demonstrate the capability of CarNet to model complex systems, we train a universal MLIP for 89 chemical elements using the MatPES r^{2}SCAN dataset[51]. We use the same data split and train for 100 epochs as in the original MatPES manuscript[51]. We trained three versions of CarNet all using N_u = 128 feature channels and a maximum natural tensor rank of L = 2, but with different numbers of GNN layers (1, 2, and 3) to investigate the accuracy–efficiency trade-off. The three-layer CarNet took ~34 hours to train on a single NVIDIA RTX 5090 GPU. Compared with state-of-the-art universal MLIPs trained on this dataset, CarNet demonstrates highly competitive performance in terms of accuracy. With only a single GNN layer, it achieves test set MAEs

Table 3 | Prediction errors in elastic properties

	K (GPa)	*G* (GPa)	*E* (GPa)	*C* (GPa)
AutoMatminer[81]	9.84	9.27	22.10	–
MatSca[6]	7.32	8.63	19.87	–
MatTen[6]	7.37	8.38	20.59	4.52
XPaiNN[50]	5.40	**6.33**	14.74	–
CarNet	**5.31**	6.39	**13.68**	**3.32**

Mean absolute errors (MAEs) of the bulk (K), shear (G), and Young's (E) moduli, as well as the 6 × 6 Voigt matrix (C) are reported. MAE of C is calculated component-wise.
Bold text indicates the smallest error.

Table 4 | Model performance on the MatPES r^{2}SCAN test set

	Energy (meV/atom)	Forces (meV/Å)	Stress (GPa)	Model size (millions)
M3GNet[52]	44	210	0.970	0.66
CHGNet[53]	30	156	0.735	2.70
TensorNet[28]	34	163	0.754	0.84
MACE[54]	25	<u>84</u>	0.711	9.06
UPET[56]	**12**	**40**	**0.201**	192.89
CarNet (1 layer)	39	175	0.897	1.60
CarNet (2 layers)	27	141	0.717	3.30
CarNet (3 layers)	<u>23</u>	130	<u>0.642</u>	7.80

Model size is measured by the number of parameters in millions. MACE and UPET are developed using data in addition to MatPES; both are pretrained on the OMat24 dataset and then finetuned on MatPES. M3GNet, CHGNet, and TensorNet results are from ref. 51; MACE and UPET results are evaluated by us using the model checkpoints released by the authors (see Data Availability).
Bold and underlined text indicate the smallest and second smallest errors, respectively.

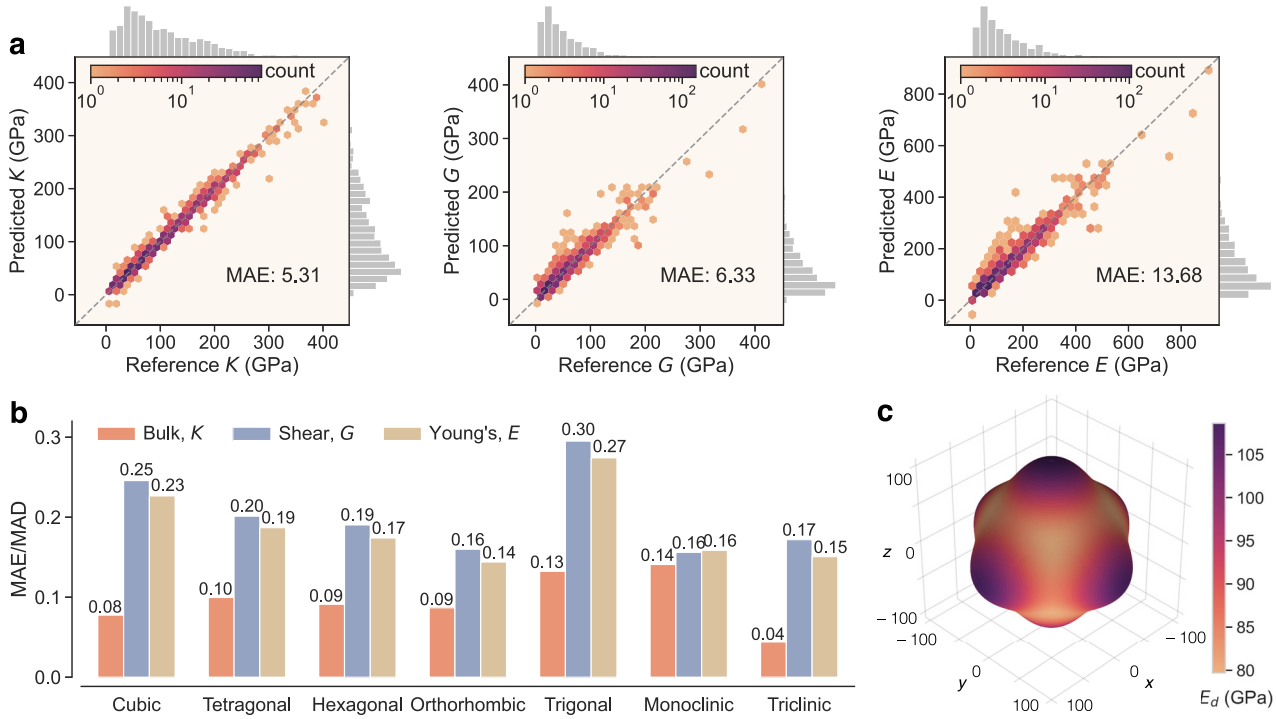

Fig. 4 | Performance of CarNet in predicting elastic properties. a Predicted bulk modulus K, shear modulus G, and Young's modulus E compared with reference density functional theory (DFT) values. **b** Normalized error by crystal system. **c** Directional Young's modulus E_d of CaS predicted by the model. The cubic symmetry of rocksalt CaS is clearly reflected in the predicted E_d. MAE is the mean absolute error, and MAD is the mean absolute deviation. Source data are provided as a Source Data file.

(Table 4) comparable to those of M3GNet[52], CHGNet[53], and TensorNet[28]. The three-layer CarNet attains overall accuracy comparable to MACE[54] (which, however, is additionally pretrained on the large OMat24 dataset[55]), with smaller energy and stress errors but larger force errors. UPET[56] achieves the smallest MAEs among all models. This advantage is most plausibly attributable to two factors: UPET employs 192.89 million parameters, more than an order of magnitude larger than any other model, and, like MACE, is pretrained on OMat24 (118 million configurations) before being finetuned on MatPES. Since MACE shares the same pretraining yet performs only on par with CarNet, UPET's additional edge appears to stem primarily from its much larger model capacity. Unlike all other models considered here, however, UPET does not guarantee rotationally invariant energy predictions, meaning that the predicted energy of a structure may change with its orientation in space. Taken together, these comparisons indicate that CarNet is highly competitive among universal MLIPs.

CarNet also demonstrates superior efficiency in terms of computational time and memory usage. We measure the inference time of the models in MD simulations of bulk water systems of different sizes. The single-layer CarNet is the fastest model of all, and the three-layer version is faster than MACE and UPET (Fig. 5a). On the NVIDIA RTX 5090 GPU with 32 GB of memory, M3GNet and all CarNet models can successfully simulate systems up to 3072 atoms (Fig. 5a), while the other models encounter out-of-memory (OOM) errors. The timing tests were also done for NaCl crystals, and similar results were obtained (see Supplementary Fig. 3). We also investigate the accuracy–efficiency trade-off by plotting the test set MAEs against the inference time for systems of 384 atoms (where all models can run without OOM errors). For both energy and forces, the CarNet models together with TensorNet and UPET form the Pareto front (Fig. 5b, c). In terms of model size (number of parameters), CarNet is an order of magnitude larger than M3GNet, CHGNet, and TensorNet, and remains slightly smaller than MACE; all are much smaller than UPET (Table 4).

Discussion

This work develops a framework for atomistic ML by leveraging Cartesian natural tensors to systematically represent high-rank and many-body interactions in atomic structures. The proposed theory of natural tensor operations enables modeling of a wide variety of physical properties, from scalar quantities (like interatomic potential energy) to tensors of arbitrary rank and symmetry (e.g., the elastic constant tensor)—a capability unmatched by existing Cartesian approaches. Key design choices enhance computational efficiency while maintaining predictive accuracy, including the implementation of sparsified tensor product paths and the optional incorporation of atomic species dependence in model parameters.

According to the product rule[34], the rank l_3 of the natural tensor $\mathbf{Z}_{l_3}$ resulting from the tensor product of $\mathbf{X}_{l_1}$ and $\mathbf{Y}_{l_2}$ satisfies $|l_1 - l_2| \le l_3 \le l_1 + l_2$. Consequently, different pairs l_1, l_2 can produce the same l_3. For example, $l_1 = 1, l_2 = 1$ and $l_1 = 1, l_2 = 2$ can both yield $l_3 = 1$. We define a path $p = (l_1, l_2, l_3)$ to denote such a combination. A central question pertains to the selection of paths that effectively contribute to the computation of atomic and hyper moments using Eq. (3). A straightforward approach, analogous to spherical models like NequIP[19], involves incorporating all possible paths, which ensures maximal expressivity but incurs significant computational cost. We find that a judicious subset of paths can often achieve superior accuracy and efficiency.

We introduce three path selection modes: 'full', 'lite', and 'level'. The full mode includes all permissible paths, whereas the lite and level modes use fewer paths by restricting the ones leading to large l_3 values. Detailed descriptions and explicit tabulations of the paths are provided in Supplementary Note 4. To evaluate their performance, we trained models with these three modes on the ethanol dataset and MatPES dataset, and checked their learning curves. For ethanol dipole moment, the lite mode yields the lowest MAE when the training data size is small (Fig. 6a, b) With a larger learning curve slope s than the

https://doi.org/10.1038/s41467-026-77263-z

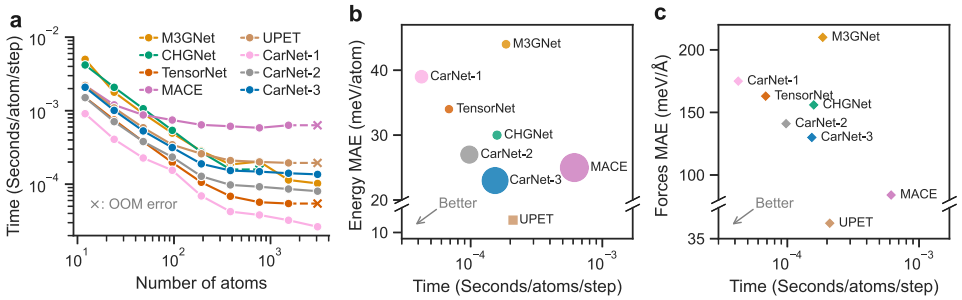

Fig. 5 | Accuracy–efficiency trade-off of various universal machine learning interatomic potentials. a Time per molecular dynamics (MD) step normalized by the number of atoms as a function of system size. **b** Test set mean absolute error (MAE) of energy versus time at the system size of 384 atoms. **c** Test set MAE of forces versus time. We measure the time by running MD simulations of bulk water systems with different sizes and report the average time per MD step normalized by the number of atoms. MD simulations were conducted using the Atomic Simulation Environment (ASE) on a single NVIDIA RTX 5090 GPU; the OOM label indicates out-of-memory errors during the calculations. CarNet-1, CarNet-2, and CarNet-3 denote the CarNet models with 1, 2, and 3 graph neural network (GNN) layers, respectively. In panel b, the marker size is proportional to the model size (number of parameters), except for UPET, whose model size is much larger than the others (see Table 4). Source data are provided as a Source Data file.

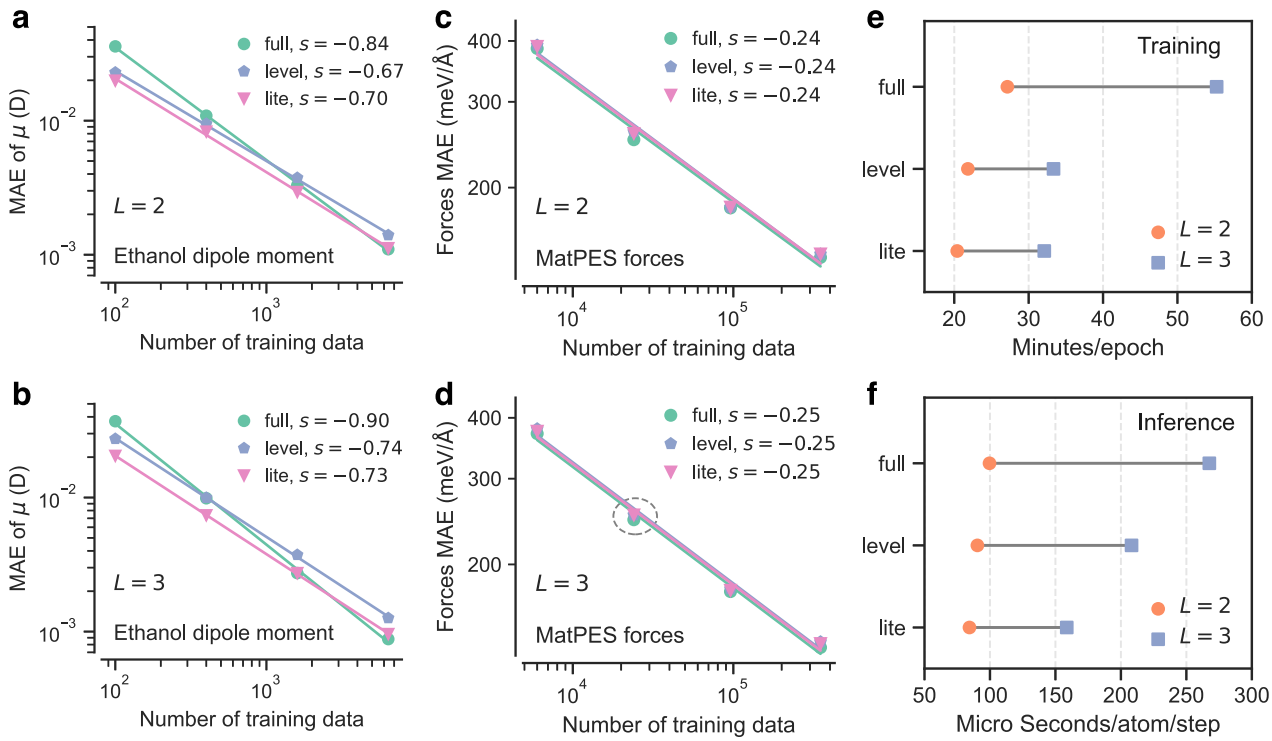

Fig. 6 | Comparison of the tensor product path selection modes. a, **b** Mean absolute error (MAE) on learning ethanol dipole moment $\boldsymbol{\mu}$ as a function of the training data size. **c**, **d** MAE of forces on the MatPES dataset as a function of the training data size. **e**, **f** Training time on the MatPES dataset and inference time using the trained models in molecular dynamics (MD) simulations of bulk water with 384 atoms. The slope s is obtained by linearly fitting the learning curve in log-log space. L denotes the maximum rank of the natural tensors used in the model. Other hyperparameters for training the ethanol dipole moment model are: number of feature channels $N_u = 64$, number of layers $T = 2$, and cutoff radius $r_{\text{cut}} = 5$ Å. Other hyperparameters for MatPES training are: $N_u = 128$, $T = 2$, and $r_{\text{cut}} = 5$ Å. Source data are provided as a Source Data file.

other two modes, the full mode's MAE decreases more rapidly with increasing training data, eventually outperforming the lite mode at larger sizes. For the MatPES dataset, the full mode consistently gives the lowest force MAEs across all training data sizes (Fig. 6c, d), although the differences between the modes are less pronounced (e.g. 247, 252, and 252 meV/Å for the full, lite, and level modes, respectively, at 24,000 training configurations, circled in Fig. 6d). Additional results on using varied training settings such as different numbers of feature channels (Supplementary Fig. 4) and on other properties such as energy and stress (Supplementary Fig. 5) show the same behavior. We also compared the efficiency of the three modes, finding that the lite mode is much faster than the full mode in both model training and inference (Fig. 6d, e). For example, when using a maximum tensor rank of $L = 2$, the lite mode and the full mode take 20.4 and 27.1 minutes per epoch, respectively, to train on the MatPES dataset; they take 84 and 99 microseconds per atom per step, respectively, to run an MD simulation of bulk water with 384 atoms. For $L = 3$ with more path reduction, the efficiency advantage of the lite mode is even larger.

The different learning-curve trends can be attributed to the interplay between the expressiveness of the path selection modes and the underlying delicacy of the datasets. The full mode with all tensor product paths possesses a higher degree of expressivity that enables it

to resolve the delicate interatomic interactions within the ethanol dataset. Specifically, it captures the subtle and fine details between molecular conformations along AIMD trajectories that the other two modes can overlook. As the training data size increases, this higher model capacity allows the full mode to more accurately map these delicate features, manifesting as a steeper learning-curve slope. Paradoxically, while the MatPES dataset exhibits greater chemical and structural complexity due to its inclusion of 89 elements and diverse crystal structures, it lacks the subtlety of the ethanol trajectories. Consequently, the additional degrees of freedom provided by the full mode remain underutilized on MatPES, leading to a saturation of performance where all three path selection modes result in nearly identical learning curves. To further validate this, we have conducted additional experiments on the relatively simple LiPS dataset and the more complex elasticity dataset. Exactly the same trends are observed (Supplementary Figs. 6, 7). This suggests that the effectiveness of high-rank tensor product paths is governed less by elemental diversity and more by the delicacy of the underlying potential energy surface. However, dataset complexity, particularly chemical diversity, determines the optimal choice of model parameterization. For datasets with a few chemical elements like ethanol and LiPS, we observe that making the weights in Eqs. (12) and (14) dependent on the atomic number z_i can enhance accuracy. This, however, does not hold for the MatPES and elasticity datasets with many chemical elements, as it leads to a large number of parameters that cannot be reliably learned from the available data.

Why Cartesian natural tensors? The field of atomistic ML has traditionally relied on spherical tensor representations and achieved considerable success. In contrast, the adoption of Cartesian representations lags behind, likely because the theoretical framework of natural tensors is less well-known and not yet fully developed. Here, we address these issues by extensions to the theory and mathematical formalism of natural tensors, as well as providing practical implementation strategies. While Cartesian and spherical representations describe the same underlying spatial reality and are mathematically connected (e.g., the tensor product $\hat{\mathbf{r}} \otimes \hat{\mathbf{r}} \otimes \cdots \otimes \hat{\mathbf{r}}$ can be expanded in terms of spherical harmonics[57]), their mathematical formulations and computational implementations differ significantly, each offering distinct advantages and limitations[26]. In addition, as discussed in Section 2.1, natural tensors possess clear physical interpretability.

There are, however, current limitations that may be further addressed to expand the potential of our approach. For example, the tensor product path selection scheme is primarily guided by empirical intuition; although different choices like the lite and level modes are comparable in the number of paths, the lite mode outperforms the level mode in general. This suggests that a more systematic approach to path selection could lead to further improvements in model performance and efficiency. Furthermore, other strategies for reducing the computational cost of the tensor product operation have also been proposed in the literature. For example, So3krates[58] achieves efficiency by decoupling invariant features from equivariant coordinates to drive an attention mechanism; eSCN[59] rotates frames into local coordinate systems to achieve a sparse representation; the Gaunt tensor product method[60] maps the Clebsch–Gordan coefficients to a 2D Fourier basis, enabling the use of Fast Fourier Transforms to reduce the complexity of full tensor products. Distinct from these approaches, the high-order pair-reduced neural network[61] avoids the tensor product completely by utilizing a hierarchical angular interaction scheme based on direct concatenation and Hadamard products. Our tensor product path selection scheme adopts a different strategy, which is to sparsify the tensor product paths. However, while these strategies were developed in the context of Clebsch–Gordan tensor products for spherical models, they may be adapted to the natural tensor product in Cartesian models. Finally, dedicated GPU kernels have been optimized for spherical tensor operations[62,63], and similar optimizations may be implemented for natural tensor operations to further improve both training and inference efficiency.

While this work focuses on atomistic ML, the proposed Cartesian natural tensor framework is broadly applicable to other domains. Atomistic ML can be viewed as a specific instance of point cloud learning, where each point (atom) possesses attributes such as atomic number and spatial coordinates. Consequently, the natural tensor formalism and the CarNet architecture can be extended to a variety of other point cloud tasks, such as shape learning in 3D medical images[64,65] and object recognition and segmentation in LiDAR data for autonomous driving applications[66,67].

Methods

Dataset

The LiPS dataset[19] consists of 250001 structures of lithium phosphorus sulfide ($Li_{6.75}P_3S_{11}$) solid-state electrolyte, generated from an AIMD trajectory. Each structure consists of 27 Li atoms, 12 P atoms, and 44 S atoms. Random subsets of 1000, 1000, and 5000 structures were selected for training, validation, and test, respectively.

The water dataset[44] consists of 1593 water configurations of 192 atoms each, obtained from AIMD simulations at 300 K. It was randomly divided into training, validation, and test sets with a split ratio of 90:5:5.

The ethanol dataset[27] includes 10,000 molecular structures, with five target properties (energy, forces, dipole moment, polarizability, and nuclear shielding), all computed in vacuum using DFT. It was randomly divided into training, validation, and test sets with a split ratio of 8:1:1.

The MatPES r^{2}SCAN dataset[51] consists of 387,897 DFT calculations of 89 elements in various crystal structures. We use the same training, validation, and test splits as in the original work[51], which are 80%, 10%, and 10% of the total dataset, respectively.

The elasticity dataset[25] contains elastic constant tensors for inorganic crystals from the DFT data in the Materials Project[68]. It includes 10276 elastic constant tensors, and we use the same training, validation, and test splits as in the original work[6], which are 80%, 10%, and 10% of the total dataset, respectively.

Model architecture

CarNet is a graph neural network (GNN) that takes an atomic structure as input and predicts target physical properties. The model comprises four key components: atomic-species embeddings; radial and directional representations of interatomic vectors using radial basis functions and natural tensors, respectively; atomic and hyper moments constructed through natural tensor products; and output heads for predicting target properties. Below, we describe each component in detail.

An atomic structure is represented as a graph $G = (V, E)$, where the nodes V correspond to atoms, and the edges E connect pairs of atoms within a cutoff radius r_{cut}. A node i (atom) is characterized by three properties: the atomic coordinates $\mathbf{r}^i$, atomic number z_i, and atom features $\mathbf{h}^i$. For each edge (i, j) we define the relative position vector as: $\mathbf{r}^{ij} = \mathbf{r}^j - \mathbf{r}^i$, which encodes the spatial relationship between atom i and its neighbor j.

The atom features $\mathbf{h}^i$ consist of a set of natural tensors, indexed by u and l. In their tensorial form, these features are represented as $\mathbf{h}^i_{ul}$, where l denotes the tensor rank and u labels the feature channels. The initial features of each atom are derived by embedding its atomic number:

$$\mathbf{h}^i_{u0} = W_{uz^i}, \tag{8}$$

where W_{uz^i} is a learnable embedding matrix. These initial atom features are scalar-valued and constitute natural tensors of rank $l = 0$.

The unit edge vector $\hat{\mathbf{r}}^{ij} = \mathbf{r}^{ij}/r^{ij}$ with $r^{ij} = \|\mathbf{r}^{ij}\|$ encodes the directional information between atoms i and j. Atom indices i and j in $\mathbf{r}^{ij}$ are omitted for simplicity. To capture angular interactions of various orders, we construct the polyadic tensor $\mathbf{U}_l = \hat{\mathbf{r}} \otimes \hat{\mathbf{r}} \otimes \cdots \otimes \hat{\mathbf{r}}$ where the tensor product is taken l times, producing a rank-l tensor. Each $\mathbf{U}_l$ can be decomposed into its natural tensor components, resulting in the set:

$$\mathbf{X}_0 = 1,\ \mathbf{X}_1 = \begin{bmatrix} \hat{r}_x \\ \hat{r}_y \\ \hat{r}_z \end{bmatrix},\ \mathbf{X}_2 = \begin{bmatrix} \hat{r}_x^2 - \sigma_0 & \hat{r}_x\hat{r}_y & \hat{r}_x\hat{r}_z \\ \hat{r}_y\hat{r}_x & \hat{r}_y^2 - \sigma_0 & \hat{r}_y\hat{r}_z \\ \hat{r}_z\hat{r}_x & \hat{r}_z\hat{r}_y & \hat{r}_z^2 - \sigma_0 \end{bmatrix} \tag{9}$$

and so forth, where $\hat{r}_x$, $\hat{r}_y$ and $\hat{r}_z$ are the Cartesian components of $\hat{\mathbf{r}}$, and $\sigma_0 = (\hat{r}_x^2 + \hat{r}_y^2 + \hat{r}_z^2)/3$, which is one-third the trace of $\mathbf{U}_2$—ensuring $\mathbf{X}_2$ is traceless (more explicitly, $\mathbf{X}_l = \mathbf{X}_l^{ij}$).

The interatomic distance (edge length) r^{ij} is expanded in a set of radial basis functions B_u indexed by channel u. Basis functions are constructed as linear combinations of Chebyshev polynomials of the first kind Q_β,

$$B_u(r) = \sum_{\beta=0}^{N_\beta} W_{u\beta} Q_\beta\left(\frac{r}{r_{\text{cut}}}\right) f_c\left(\frac{r}{r_{\text{cut}}}\right), \tag{10}$$

where N_β is the maximum degree of the Chebyshev polynomials and $W_{u\beta}$ are learnable weights. The radial expansion is similar to that used in MTP[69] and CAMP[25], but here the weights $W_{u\beta}$ are shared across different atomic species, significantly reducing the total number of learnable parameters; this is advantageous for datasets with many chemical elements.

Using the atom features, angular information, and radial basis, we construct an atomic moment that encodes the local environment of each atom:

$$\mathbf{M}^i_{ul_3,p} = \frac{1}{\sqrt{|\mathcal{N}|}} \sum_{j\in\mathcal{N}_i} R_{ul_3l_1l_2} \mathbf{h}^j_{ul_1} \hat{\otimes} \mathbf{X}^{ij}_{l_2}, \tag{11}$$

where $\mathcal{N}_i$ is the set of neighboring atoms within a distance of r_{cut} of atom i, $|\mathcal{N}|$ is the average number of neighbors per atom in the training set, $R_{ul_3l_1l_2}$ is a learnable radial function, and $\hat{\otimes}$ denotes the natural tensor product (as defined in Section 2.1). The radial function $R_{ul_3l_1l_2}$ is obtained by passing the radial basis B_u through a multilayer perceptron (MLP): $R_{ul_3l_1l_2} = \text{MLP}(\text{B}_\text{u})$ using two hidden layers with the SiLU nonlinearity[70]. Different MLPs are used for each combination of indices (l_3, l_2, l_1). Similar to spherical tensor products, the product of natural tensors of ranks l_1 and l_2 can produce a natural tensor of rank l_3. The triplet $p = (l_1, l_2, l_3)$ is called a path and determines the tensorial combination in Eq. (11).

Atomic moment tensors of the same rank but originating from different tensorial paths are linearly combined as follows:

$$\mathbf{M}^i_{ul} = \sum_{u'p} W^{z_i}_{uu'l,p} \mathbf{M}^i_{u'l,p}, \tag{12}$$

where $W^{z_i}_{uu'l,p}$ are trainable weights. To reduce the number of parameters, these weights are factored as $W^{z_i}_{uu'l,p} = W^{z_i}_{uu'l} W_p$.

From the atomic moments, we construct the hyper moment,

$$\mathbf{H}^i_{ul} = \mathbf{M}^i_{ul_1} \hat{\otimes} \mathbf{M}^i_{ul_2} \hat{\otimes} \cdots \hat{\otimes} \mathbf{M}^i_{ul_\upsilon} \quad (\upsilon \text{ of } \mathbf{M}). \tag{13}$$

where $\hat{\otimes}$ denotes the natural tensor product and υ is the number of atomic moments being combined. An atomic moment encodes two-body interactions between an atom and its neighbors. By taking tensor product of atomic moments with themselves, higher-order interactions are incorporated: three-body, four-body, and beyond. Specifically, a hyper moment of degree υ captures interactions up to body order $\upsilon + 1$. The hyper moment thus provides a systematic and complete description of the local atomic environment[22,57], which is essential for constructing systematically improvable interatomic potentials. This approach is analogous to the B-basis used in ACE[17] and MACE[20]. Similar to atomic moments, multiple tensor paths can generate hyper moments of the same rank l that may be linearly combined. An efficient algorithm for evaluating Eq. (13) iteratively is provided in Supplementary Note 5.

Atom features are then updated using a residual connection[71] by combining the hyper moments with atom features of the previous layer:

$$\mathbf{h}^{i,t}_{ul} = \mathbf{H}^i_{ul} + \sum_{u'} W^{z_i}_{uu'l} \mathbf{h}^{i,t-1}_{u'l}, \tag{14}$$

where t is the layer index.

The feature update process is performed for N_{layer} layers, producing a sequence of atom features: $\mathbf{h}^{i,1}_{ul}$, $\mathbf{h}^{i,2}_{ul}$, ..., $\mathbf{h}^{i,N_{\text{layer}}}_{ul}$. Depending on the modeling target, the features from either all layers or a subset of layers are used to construct the final output. Empirically, two or three layers are sufficient for interatomic potentials, atomic tensors, and structure tensors.

For interatomic potentials, the atomic energy E^i is derived from $l = 0$ scalar atom features $\mathbf{h}^{i,t}_{u0}$ across all layers

$$E^i = \sum_{t=1}^{N_{\text{layer}}} V(\mathbf{h}^{i,t}_{u0}), \tag{15}$$

in which V is implemented as an MLP using two hidden layers with the SiLU nonlinearity for the last layer where $t = N_{\text{layer}}$, and a linear function, $V(\mathbf{h}^{i,t}_{u0}) = \sum_u W^t_u \mathbf{h}^{i,t}_{u0}$, for earlier layers ($t < N_{\text{layer}}$). The total potential energy is the sum over all atoms:

$$E = \sum_i \sigma E^i + \mu_{z_i}, \tag{16}$$

where μ_{z_i} is the atomic energy for species z_i (obtained directly from DFT calculations or computed via a linear fit to the total energies of the training set), and σ is the root mean square of the atomic forces computed on the training set[54,72]. Forces are obtained via the negative gradient: $\mathbf{F}_i = -\frac{\partial E}{\partial \mathbf{r}_i}$.

For atomic tensors, the atom features $\mathbf{h}^{i,1}_{ul}$, $\mathbf{h}^{i,2}_{ul}$, ..., $\mathbf{h}^{i,N_{\text{layer}}}_{ul}$ from all layers are linearly combined along the channel dimension u to produce a rank-l natural tensor for each atom:

$$\mathbf{v}^i_l = \sum_{t,u} W^t_u \mathbf{h}^{i,t}_{ul}. \tag{17}$$

Relevant natural tensors are then selected to reconstruct physical tensors.

For example, the rank-0 $\mathbf{v}^i_0$, rank-1 $\mathbf{v}^i_1$ and rank-2 $\mathbf{v}^i_2$ natural tensors are used to reconstruct the rank-2 nuclear shielding tensor $\boldsymbol{\sigma}$ for each atom i. For the shielding tensor, there is only one unique independent mapping tensor for each rank, i.e., $N_g = 1$ for all m in Eq. (7). Therefore, the reconstruction of the shielding tensor is given by:

$$\boldsymbol{\sigma}^i = \sum_{m=0}^{2} \mathbf{Q}_{m+2} \odot^m \mathbf{X}_m \tag{18}$$

where $\mathbf{X}_0 = \mathbf{v}^i_0$, $\mathbf{X}_1 = \mathbf{v}^i_1$, and $\mathbf{X}_2 = \mathbf{v}^i_2$ are the natural tensors of rank 0, 1, and 2, respectively.

For structural tensors, natural tensors of different ranks l are first generated for each atom according to Eq. (17). They are then aggregated across all atoms by taking the average or sum, depending on whether the target physical tensor is intensive or extensive. Then relevant natural tensors are selected to reconstruct the physical tensor.

For extensive quantities (e.g., the dipole moment $\mathbf{u}$ and polarizability $\boldsymbol{\alpha}$), the tensors are summed:

$$\mathbf{X}_l = \sum_i \mathbf{v}_l^i. \tag{19}$$

According to the decomposition and recombination spectrum (given in Supplementary Table 3), the dipole moment $\mathbf{u}$ only needs the rank-1 natural tensor $\mathbf{X}_1$ for its reconstruction. It can be obtained via Eq. (7) as:

$$\mathbf{u} = \mathbf{Q}_{m+1} \odot^m \mathbf{X}_m, \tag{20}$$

where $m = 1$. The polarizability $\boldsymbol{\alpha}$ is a symmetric rank-2 tensor, which can be reconstructed from the rank-0 and rank-2 natural tensors ($\mathbf{X}_0$ and $\mathbf{X}_2$) as:

$$\boldsymbol{\alpha} = \sum_{m \in \{0,2\}} \mathbf{Q}_{m+2} \odot^m \mathbf{X}_m. \tag{21}$$

For the intensive elastic constant tensor $\mathbf{C}$, the average is taken:

$$\mathbf{X}_l = \frac{1}{N} \sum_i \mathbf{v}_l^i, \tag{22}$$

where N is the total number of atoms in the structure. The rank-4 elastic constant tensor $\mathbf{C}$ is more complex: it decomposes into two rank-0 natural tensors ($\mathbf{X}_0^1$ and $\mathbf{X}_0^2$), two rank-2 natural tensors ($\mathbf{X}_2^1$ and $\mathbf{X}_2^2$), and a rank-4 natural tensor $\mathbf{X}_4$ (see Supplementary Table 3). The two tensors $\mathbf{X}_0^1$ and $\mathbf{X}_0^2$ of the same rank are obtained by employing separate W_u^t in Eq. (17); similarly for $\mathbf{X}_2^1$ and $\mathbf{X}_2^2$. Then the elastic constant tensor is reconstructed via Eq. (7) as:

$$\mathbf{C} = \sum_{m \in \{0,2,4\}} \sum_{g=1}^{N_g(m)} \mathbf{Q}_{m+4}^g \odot^m \mathbf{X}_m^g, \tag{23}$$

where $N_g(0) = 2$, $N_g(2) = 2$, and $N_g(4) = 1$.

Normalization

Properly normalizing the internal features of a neural network is critical for ensuring stable and efficient training. In this work, we adopt the default uniform initialization scheme in PyTorch[73] for all learnable weights, but we update the initialization bounds so that all network components yield outputs with approximately zero mean and unit variance. Furthermore, when summing over the features of neighboring atoms to compute the atomic moment in Eq. (11), we normalize the sum by dividing by the square root of the average number of neighbors, $\sqrt{|\mathcal{N}|}$, in the training set. This normalization assumes that contributions from different terms are uncorrelated, such that their variances are additive[72].

Normalizing learning targets is equally crucial for model performance[72]. This normalization process is equivalent to the choice of scale and shift parameters for the model's output. For interatomic potentials, we use the root mean square of the atomic forces σ as the scale parameter and the atomic energy μ_{z_i} as the shift parameter, as described in Eq. (16). For tensorial properties, we perform normalization in the natural tensor space rather than the Cartesian physical tensor space, using rank-dependent scale and shift parameters. For scalar natural tensors ($l = 0$), the shift μ_l and scale σ_l are set to the mean and standard deviation of the target scalar values across the training set. For higher-rank natural tensors ($l > 0$), the shift is set to zero ($\mu_l = 0$) to maintain equivariance, while the scale σ_l is set to the root mean square of the Frobenius norm of the target natural tensors of the same rank. Applying these parameters to the atomic natural tensors, Eq. (17) becomes: $\mathbf{v}_l^i = \sigma_l \left(\sum_{t,u} W_u^t \mathbf{h}_{ul}^{i,t} \right) + \mu_l$. We note that although this normalization is conducted at the atomic level, the formulation remains valid for structural targets, which are simply linear combinations of atomic values. For example, under this normalization, Eq. (22) becomes: $\mathbf{v}_l = \frac{1}{N} \sum_i (\sigma_l \mathbf{v}_l^i + \mu_l) = \sigma_l (\frac{1}{N} \sum_i \mathbf{v}_l^i) + \mu_l$, which is identical to normalizing the structural tensor directly.

Model training

Interatomic potentials are trained by minimizing a loss function of energy and forces (and stress if available). For a given atomic structure, the loss is

$$l(\theta) = w_\mathcal{E} \left(\frac{\mathcal{E} - \hat{\mathcal{E}}}{N} \right)^2 + w_\mathrm{F} \frac{\sum_{i=1}^N \| \mathbf{F}_i - \hat{\mathbf{F}}_i \|^2}{3N} + w_\mathrm{S} \| \mathbf{S} - \hat{\mathbf{S}} \|^2, \tag{24}$$

where N is the number of atoms, $\mathcal{E}$, $\mathbf{F}_i$, and $\mathbf{S}$ are the model-predicted energy, forces, and stress, $\hat{\mathcal{E}}$, $\hat{\mathbf{F}}_i$, and $\hat{\mathbf{S}}$ are the corresponding reference values, and $w_\mathcal{E}$, w_F, and w_S are the energy, force, and stress weighting factors.

For atomic tensor properties (e.g., nuclear shielding tensor), the loss is computed as

$$l(\theta) = \frac{1}{N} \sum_{i=1}^N \| \boldsymbol{\sigma}^i - \hat{\boldsymbol{\sigma}}^i \|^2, \tag{25}$$

where $\boldsymbol{\sigma}^i$ is the predicted Cartesian rank-2 nuclear shielding tensor for atom i, and $\hat{\boldsymbol{\sigma}}^i$ is its reference value. For structural tensor properties (e.g., dipole moment, polarizability, and elastic constant tensors), the loss per structure is

$$l(\theta) = \| \mathbf{T} - \hat{\mathbf{T}} \|^2, \tag{26}$$

where $\mathbf{T}$ and $\hat{\mathbf{T}}$ are the predicted and reference tensors. The total loss to minimize at each optimization step is the mean of the losses of a minibatch of configurations.

For the elastic constant tensor dataset and the MatPES dataset, we also used the Huber loss[74] for training, which is less sensitive to outliers than the mean squared error loss. It is defined as

$$l_\delta = \begin{cases} \frac{1}{2} \| y - \hat{y} \|^2, & \text{for } \| y - \hat{y} \| < \delta \\ \delta \| y - \hat{y} \| - \frac{1}{2}\delta^2, & \text{otherwise} \end{cases}, \tag{27}$$

where y and $\hat{y}$ denote the predicted and reference values, respectively (e.g., $\mathbf{T}$ and $\hat{\mathbf{T}}$ for the elastic constant tensor), and δ is a hyperparameter that determines the threshold for switching between the quadratic and linear loss regimes.

The models are implemented in PyTorch[73] and trained using PyTorch Lightning[75]. Optimization is performed with the AdamW optimizer[76]. A cosine annealing learning rate schedule is employed, starting with an initial learning rate depending on the specific task. During model performance evaluation, an exponential moving average of the model weights is maintained, with decay rates of 0.999 for interatomic potential models and 0.99 for tensor property models. Hyperparameters are selected based on model performance on the validation set, and all results correspond to the test set.

Detailed model training hyperparameters for all tasks are provided in Supplementary Table 1.

Molecular dynamics

MD simulations are performed in the NVT ensemble using the Nosé–Hoover thermostat. For LiPS, the simulation employs a cell size identical to that used in the training data, consisting of 83 atoms. The system is maintained at 520 K with a timestep of 1 fs and a Nosé–Hoover chain damping time of 20 fs. The total simulation duration is 300 ps. The thermostat is implemented with the ASE `NoseHooverChain`[77], using a single chain. Simulations for water are similarly conducted using a cell with 512 molecules at temperatures from 300 to 2000 K.

The diffusion coefficient D is calculated from the mean squared displacement (MSD) using the Einstein relation[78]:

$$D = \lim_{t\to\infty} \frac{\left\langle \frac{1}{N}\sum_i^N |\mathbf{r}_i(t) - \mathbf{r}_i(0)|^2 \right\rangle}{2nt}, \tag{28}$$

where N is the number of diffusing atoms (lithium for LiPS, oxygen for water), $\mathbf{r}_i(t)$ is the position of atom i at time t, $\langle \cdot \rangle$ denotes an ensemble average over multiple time origins or trajectories, and $n = 3$ indicates diffusion in three dimensions. In practice, the diffusion coefficient is computed from the slope of a linear fit to the MSD versus $2nt$, as implemented in ASE[77]. The data from the initial 10 ps is discarded; the remaining data is used in the fitting (Fig. 3c, f). This ensures an accurate estimation of the diffusion coefficient from the long-time linear regime of the MSD.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

Source data are provided with this paper. All datasets used to train models are publicly available: The LiPS dataset is at https://archive.materialscloud.org/record/2022.45, the water dataset is at https://doi.org/10.1073/pnas.1815117116, the ethanol dataset is at http://quantum-machine.org, the elasticity dataset is at https://doi.org/10.5281/zenodo.8190849, and the MatPES dataset (v2025.1) is at https://matpes.ai. The CarNet models generated in this study have been deposited in a GitHub repository at https://github.com/wengroup/carnet_run. The MACE and UPET models trained on the OMAT dataset and finetuned on the MatPES dataset are publicly available. The 'MACE-matpes-r2scan-omat-ft.model' checkpoint is at https://github.com/ACEsuit/mace-foundations/releases/tag/mace_matpes_0, and the 'pet-omatpes-l-v0.1.0.ckpt' checkpoint is at https://huggingface.co/lab-cosmo/upet/tree/main/models. Source data are provided with this paper.

Code availability

The CarNet source code is at https://github.com/wengroup/carnet; the version used to generate the results reported in this work, commit `186071c81b49cd1d254c751ed4a4f6b8f1b85df9`, is archived at Zenodo[79]. The code for natural tensor operations is at https://github.com/wengroup/natt. Scripts for training models, running MD simulations, and analyzing the results are at https://github.com/wengroup/carnet_run.

References

1. Unke, O. T. et al. Machine learning force fields. *Chem. Rev.* **121**, 10142–10186, https://doi.org/10.1021/acs.chemrev.0c01111 (2021).
2. Musil, F. et al. Physics-inspired structural representations for molecules and materials. *Chem. Rev.* **121**, 9759–9815, https://doi.org/10.1021/acs.chemrev.1c00021 (2021).
3. Wen, M., Blau, S. M., Spotte-Smith, E. W. C., Dwaraknath, S. & Persson, K. A. Bondnet: a graph neural network for the prediction of bond dissociation energies for charged molecules. *Chem. Sci.* **12**, 1858–1868, https://doi.org/10.1039/D0SC05251E (2020).
4. Ruff, R., Reiser, P., Stühmer, J. & Friederich, P. Connectivity optimized nested line graph networks for crystal structures. *Digital Discov.* **3**, 594–601, https://doi.org/10.1039/D4DD00018H (2024).
5. Veit, M., Wilkins, D. M., Yang, Y., DiStasio, R. A. & Ceriotti, M. Predicting molecular dipole moments by combining atomic partial charges and atomic dipoles. *J. Chem. Phys.* **153**, https://doi.org/10.1063/5.0009106 (2020).
6. Wen, M., Horton, M. K., Munro, J. M., Huck, P. & Persson, K. A. An equivariant graph neural network for the elasticity tensors of all seven crystal systems. *Digital Discov.* **3**, 869–882, https://doi.org/10.1039/D3DD00233K (2024).
7. Yao, Z. et al. Machine learning for a sustainable energy future. *Nat. Rev. Mater.* **8**, 202–215, https://doi.org/10.1038/s41578-022-00490-5 (2023).
8. Wen, M. et al. Chemical reaction networks and opportunities for machine learning. *Nat. Comput. Sci.* **3**, 12–24, https://doi.org/10.1038/s43588-022-00369-z (2023).
9. Li, H., Jiao, Y., Davey, K. & Qiao, S.-Z. Data-driven machine learning for understanding surface structures of heterogeneous catalysts. *Angew. Chem. Int. Ed.* **62**, e202216383, https://doi.org/10.1002/ange.202216383 (2023).
10. Mou, T. et al. Bridging the complexity gap in computational heterogeneous catalysis with machine learning. *Nat. Catal.* **6**, 122–136, https://doi.org/10.1038/s41929-023-00911-w (2023).
11. Vamathevan, J. et al. Applications of machine learning in drug discovery and development. *Nat. Rev. Drug Discov.* **18**, 463–477, https://doi.org/10.1038/s41573-019-0024-5 (2019).
12. Ekins, S. et al. Exploiting machine learning for end-to-end drug discovery and development. *Nat. Mater.* **18**, 435–441, https://doi.org/10.1038/s41563-019-0338-z (2019).
13. Behler, J. & Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **98**, 146401, https://doi.org/10.1103/PhysRevLett.98.146401 (2007).
14. Rupp, M., Tkatchenko, A., Müller, K.-R. & Von Lilienfeld, O. A. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.* **108**, 058301, https://doi.org/10.1103/PhysRevLett.108.058301 (2012).
15. Zhang, L., Han, J., Wang, H., Car, R. & E, W. Deep potential molecular dynamics: A scalable model with the accuracy of quantum mechanics. *Phys. Rev. Lett.* **120**, 143001, https://doi.org/10.1103/PhysRevLett.120.143001 (2018).
16. Thompson, A. P., Swiler, L. P., Trott, C. R., Foiles, S. M. & Tucker, G. J. Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials. *J. Comput. Phys.* **285**, 316–330, https://doi.org/10.1016/j.jcp.2014.12.018 (2015).
17. Drautz, R. Atomic cluster expansion for accurate and transferable interatomic potentials. *Phys. Rev. B* **99**, 014104, https://doi.org/10.1103/PhysRevB.99.014104 (2019).
18. Thomas, N. et al. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. Preprint at https://doi.org/10.48550/arXiv.1802.08219 (2018).
19. Batzner, S. et al. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nat. Commun.* **13**, 1–11, https://doi.org/10.1038/s41467-022-29939-5 (2022).
20. Batatia, I., Kovacs, D. P., Simm, G., Ortner, C. & Csányi, G. Mace: higher order equivariant message passing neural networks for fast and accurate force fields. *Adv. Neural Inf. Process. Syst.* **35**, 11423–11436, https://doi.org/10.52202/068431-0830 (2022).
21. Bochkarev, A., Lysogorskiy, Y. & Drautz, R. Graph atomic cluster expansion for semilocal interactions beyond equivariant message passing. *Phys. Rev. X* **14**, 021036, https://doi.org/10.1103/PhysRevX.14.021036 (2024).

22. Shapeev, A. V. Moment tensor potentials: a class of systematically improvable interatomic potentials. *Multisc. Model. Simula.* https://doi.org/10.1137/15M1054183 (2016).
23. Zhang, Y., Xia, J. & Jiang, B. Physically motivated recursively embedded atom neural networks: incorporating local completeness and nonlocality. *Phys. Rev. Lett.* **127**, 156002, https://doi.org/10.1103/PhysRevLett.127.156002 (2021).
24. Cheng, B. Cartesian atomic cluster expansion for machine learning interatomic potentials. *npj Comput. Mater.* **10**, 1–10, https://doi.org/10.1038/s41524-024-01332-4 (2024).
25. Wen, M., Huang, W.-F., Dai, J. & Adhikari, S. Cartesian atomic moment machine learning interatomic potentials. *npj Comput. Mater.* **11**, 128. https://doi.org/10.1038/s41524-025-01623-4 (2025).
26. Zaverkin, V. et al. Higher-rank irreducible Cartesian tensors for equivariant message passing. *Adv. Neural Inf. Process. Syst.* **37**, 124025–124068, https://doi.org/10.52202/079017-3940 (2024).
27. Gastegger, M., Schütt, K. T. & Müller, K.-R. Machine learning of solvent effects on molecular spectra and reactions. *Chem. Sci.* **12**, 11473–11483, https://doi.org/10.1039/d1sc02742e (2021).
28. Simeon, G. & De Fabritiis, G. in *Tensornet: Cartesian tensor representations for efficient learning of molecular potentials* (eds Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M. & Levine, S.) *Advances in Neural Information Processing Systems*, Vol. 36, 37334–37353 https://papers.nips.cc/paper_files/paper/2023/hash/75c2ec5f98d7b2f50ad68033d2c07086-Abstract-Conference.html. https://doi.org/10.52202/075280-1623 (2023).
29. Wang, J. et al. E(n)-equivariant Cartesian tensor message passing interatomic potential. *Nat. Commun.* **15**, 7607. https://doi.org/10.1038/s41467-024-51886-6 (2024).
30. Xu, Z., Xie, W. & Hu, P. Spectral/spatial tensor atomic cluster expansion with universal embeddings in Cartesian space. Preprint at https://arxiv.org/abs/2509.14961. https://doi.org/10.48550/arXiv.2509.14961 (2025).
31. Kundu, P., Cohen, I., Dowling, D. & Capecelatro, J. *Fluid Mechanics* (Academic Press, 2024). https://books.google.com/books?id=OrL6EAAAQBAJ.
32. Jones, D., Sneddon, I., Ulam, S. & Stark, M. *The Theory of Electromagnetism* (Pergamon, 2013). https://books.google.com/books?id=9aBGBQAAQBAJ.
33. Coope, J. A. R., Snider, R. F. & McCourt, F. R. Irreducible Cartesian tensors. *J. Chem. Phys.* **43**, 2269–2275, https://doi.org/10.1063/1.1697123 (1965).
34. Coope, J. A. R. & Snider, R. F. Irreducible Cartesian tensors. ii. General formulation. *J. Math. Phys.* **11**, 1003–1017, https://doi.org/10.1063/1.1665190 (1970).
35. Coope, J. A. R. Irreducible Cartesian tensors. iii. Clebsch-Gordan reduction. *J. Math. Phys.* **11**, 1591–1612, https://doi.org/10.1063/1.1665301 (1970).
36. Zee, A. *Group Theory in a Nutshell for Physicists* (Princeton University Press, 2016). https://press.princeton.edu/books/hardcover/9780691162690/group-theory-in-a-nutshell-for-physicists.
37. Jerphagnon, J., Chemla, D. & Bonneville, R. The description of the physical properties of condensed matter using irreducible tensors. *Adv. Phys.* **27**, 609–650, https://doi.org/10.1080/00018737800101454 (1978).
38. Lehman, D. R. & Parke, W. C. Angular reduction in multiparticle matrix elements. *J. Math. Phys.* **30**, 2797–2806, https://doi.org/10.1063/1.528515 (1989).
39. Edmonds, A. *Angular Momentum in Quantum Mechanics* Investigations in Physics Series (Princeton University Press, 1996).
40. Morris, P. R. Averaging fourth-rank tensors with weight functions. *J. Appl. Phys.* **40**, 447–448, https://doi.org/10.1063/1.1657417 (1969).
41. Harris, R. A., McClain, W. M. & Sloane, C. F. On the theory of polarized light scattering from dilute polymer solutions. *Mol. Phys.* **28**, 381–398, https://doi.org/10.1080/00268977400102921 (1974).
42. Russ, M. C. & Hollingsworth, C. A. Properties of three-dimensional Cartesian tensors. i. Some properties of irreducible tensors. *J. Math. Phys.* **23**, 1025–1027, https://doi.org/10.1063/1.525490 (1982).
43. Golub, G. & Van Loan, C. *Matrix Computations* Johns Hopkins Studies in the Mathematical Sciences (Johns Hopkins University Press, 1996). https://doi.org/10.56021/9781421407944.
44. Cheng, B., Engel, E. A., Behler, J., Dellago, C. & Ceriotti, M. Ab initio thermodynamics of liquid and solid water. *Proc. Natl. Acad. Sci. USA* **116**, 1110–1115, https://doi.org/10.1073/pnas.1815117116 (2019).
45. Fu, X. et al. Forces are not enough: benchmark and critical evaluation for machine learning force fields with molecular simulations. *Trans. Mach. Learn. Res.* https://openreview.net/forum?id=A8pqQipwkt. https://doi.org/10.48550/arXiv.2210.07237 (2023).
46. Skinner, L. B., Benmore, C. J., Neuefeind, J. C. & Parise, J. B. The structure of water around the compressibility minimum. *J. Chem. Phys.* **141**, 214507, https://doi.org/10.1063/1.4902412 (2014).
47. Soper, A., Bruni, F. & Ricci, M. Site–site pair correlation functions of water from 25 to 400 c: revised analysis of new and old diffraction data. *J. Chem. Phys.* **106**, 247–254, https://doi.org/10.1063/1.473030 (1997).
48. Voigt, W. *Lehrbuch der kristallphysik (mit Ausschluss der Kristalloptik)* (Vieweg+Teubner Verlag Wiesbaden, 1966). https://doi.org/10.1007/978-3-663-15884-4.
49. Hill, R. The elastic behaviour of a crystalline aggregate. *Proc. Phys. Soc. Sect. A* **65**, 349, https://doi.org/10.1088/0370-1298/65/5/307 (1952).
50. Yan, W. et al. General framework for geometric deep learning on tensorial properties of molecules and crystals. *J. Am. Chem. Soc.* **147**, 47044–47056, https://doi.org/10.1021/jacs.5c12428 (2025).
51. Kaplan, A. D. et al. A foundational potential energy surface dataset for materials. Preprint at https://doi.org/10.48550/arXiv.2503.04070 (2025).
52. Chen, C. & Ong, S. P. A universal graph deep learning interatomic potential for the periodic table. *Nat. Comput. Sci.* **2**, 718–728, https://doi.org/10.1038/s43588-022-00349-3 (2022).
53. Deng, B. et al. Chgnet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nat. Mach. Intell.* **5**, 1031–1041, https://doi.org/10.1038/s42256-023-00716-3 (2023).
54. Batatia, I. et al. A foundation model for atomistic materials chemistry. *J. Chem. Phys.* **163**, 184110, https://doi.org/10.1063/5.0297006 (2025).
55. Barroso-Luque, L. et al. The open materials 2024 (omat24) inorganic materials dataset and models. *Nat. Comput. Sci.* **6**, 642–652, https://doi.org/10.1038/s43588-026-00996-w (2026).
56. Mazitov, A. et al. Pet-mad as a lightweight universal interatomic potential for advanced materials modeling. *Nat. Commun.* **16**, 10653. https://doi.org/10.1038/s41467-025-65662-7 (2025).
57. Dusson, G. et al. Atomic cluster expansion: completeness, efficiency and stability. *J. Comput. Phys.* **454**, 110946, https://doi.org/10.1016/j.jcp.2022.110946 (2022).
58. Frank, J. T., Unke, O. T. & Müller, K.-R. So3krates: equivariant attention for interactions on arbitrary length-scales in molecular systems. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22 (Curran Associates Inc., 2022). https://doi.org/10.52202/068431-2132.
59. Passaro, S. & Zitnick, C. L.Reducing SO(3) convolutions to SO(2) for efficient equivariant GNNs. In *Proceedings of the 40th International Conference on Machine Learning*, Vol. 202 of *Proceedings of Machine Learning Research*, (eds Krause, A., et al.) 27420–27438 (PMLR, 2023). https://proceedings.mlr.press/v202/passaro23a.html.
60. Luo, S., Chen, T. & Krishnapriyan, A. Enabling efficient equivariant operations in the fourier basis via gaunt tensor products. In *International Conference on Learning Representations*, Vol. 2024, (eds

Kim, B. et al.) 24742–24777. https://doi.org/10.48550/arXiv.2401.10216 (2024).
61. Yang, Z.-X. et al. High-order pair-reduced neural network architecture for global potential energy surface exploration across the periodic table. *Sci. China Chem*. https://doi.org/10.1007/s11426-025-3054-y (2025).
62. Cuequivariance library. GitHub https://github.com/NVIDIA/cuEquivariance (2025).
63. Bharadwaj, V., Glover, A., Buluc, A. & Demmel, J. An efficient sparse kernel generator for o(3)-equivariant deep networks. In *SIAM Conference on Applied and Computational Discrete Algorithms (ACDA25)* (Society for Industrial and Applied Mathematics, 2025). https://doi.org/10.1137/1.9781611978759.3.
64. Chen, X. et al. Shape registration with learned deformations for 3d shape reconstruction from sparse and incomplete point clouds. *Med. Image Anal.* **74**, 102228, https://doi.org/10.1016/j.media.2021.102228 (2021).
65. Zhang, G., Yang, J. & Li, Y.*Hierarchical feature learning for medical point clouds via state space model*. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2025*, Lecture Notes in Computer Science, 256–265 (Springer Nature Switzerland, 2025). https://doi.org/10.1007/978-3-032-05127-1_25.
66. Qi, C. R., Su, H., Mo, K. & Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* https://doi.org/10.1109/cvpr.2017.16 (2017).
67. Shi, W. & Rajkumar, R. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* https://doi.org/10.1109/CVPR42600.2020.00178 (2020).
68. Horton, M. K. et al. Accelerated data-driven materials science with the materials project. *Nat. Mater*. https://doi.org/10.1038/s41563-025-02272-0 (2025).
69. Novikov, I. S., Gubaev, K., Podryabinkin, E. V. & Shapeev, A. V. The mlip package: moment tensor potentials with mpi and active learning. *Mach. Learn. Sci. Technol.* **2**, 025002, https://doi.org/10.1088/2632-2153/abc9fe (2020).
70. Elfwing, S., Uchibe, E. & Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Netw.* **107**, 3–11, https://doi.org/10.1016/j.neunet.2017.12.012 (2018).
71. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778 (IEEE, 2016). https://doi.org/10.1109/CVPR.2016.90.
72. Musaelian, A. et al. Learning local equivariant representations for large-scale atomistic dynamics. *Nat. Commun.* **14**, 1–15, https://doi.org/10.1038/s41467-023-36329-y (2023).
73. Paszke, A. et al. Pytorch: an imperative style, high-performance deep learning library. Preprint at https://doi.org/10.48550/arXiv.1912.01703 (2019).
74. Huber, P. J. Robust estimation of a location parameter. *Ann. Math. Stat.* **35**, 73–101, https://doi.org/10.1214/aoms/1177703732 (1964).
75. Pytorch-lightning. GitHub https://github.com/Lightning-AI/pytorch-lightning (2025).
76. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. Preprint at https://doi.org/10.48550/arXiv.1412.6980 (2014).
77. Larsen, A. H. et al. The atomic simulation environment—a python library for working with atoms. *J. Phys. Condens. Matter* **29**, 273002, https://doi.org/10.1088/1361-648X/aa680e (2017).
78. Einstein, A. Über die von der molekularkinetischen theorie der wärme geforderte bewegung von in ruhenden flüssigkeiten suspendierten teilchen. *Ann. Der Phys.* **322**, 549–560, https://doi.org/10.1002/andp.19053220806 (1905).
79. Chen, Q., Pattamatta, A. S. L. S., Wang, B., Srolovitz, D. J. & Wen, M. Atomistic machine learning with irreducible Cartesian natural tensors (this paper). *Zenodo* https://doi.org/10.5281/zenodo.21098816 (2026).
80. Schütt, K., Unke, O. & Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *International Conference on Machine Learning*, Vol. 139, 9377–9388 (PMLR, 2021). https://doi.org/10.48550/arXiv.2102.03150.
81. Dunn, A., Wang, Q., Ganose, A., Dopp, D. & Jain, A. Benchmarking materials property prediction methods: the matbench test set and automatminer reference algorithm. *npj Comput. Mater.* **6**, 138. https://doi.org/10.1038/s41524-020-00406-3 (2020).
82. Pattamatta, A. S. L. S. A blender-powered toolkit for scientific visualization of atoms, molecules, crystals, and beyond. *GitHub* https://github.com/spattamatta/atomviz (2025).

Acknowledgements

This work is supported by startup funding from the University of Electronic Science and Technology of China (UESTC) and uses computational resources provided by the Center for High-Performance Computing (HPC) at UESTC. It also uses computational resources provided by the Hefei Advanced Computing Center.

Author contributions

Q.C.: model training, data analysis, and writing - review; A.S.L.S.P.: data analysis, visualization, and writing - review; B.W.: data analysis and writing - review; D.J.S.: writing - review; M.W.: project conceptualization, model development, data analysis, visualization, writing - original draft, writing - review, and supervision.

Funding

M.W. discloses support for the research of this work from startup funding awarded by the University of Electronic Science and Technology of China. Q.C., A.S.L.S.P., B.W., and D.J.S. declare no relevant funding.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-026-77263-z.

Correspondence and requests for materials should be addressed to Mingjian Wen.

Peer review information *Nature Communications* thanks anonymous reviewers for their contribution to the peer review of this work. A peer review file is available.