nature communications

Article

https://doi.org/10.1038/s41467-023-36329-y

Learning local equivariant representations for large-scale atomistic dynamics

Received: 16 June 2022

Accepted: 23 January 2023

Published online: 03 February 2023

Check for updates

Albert Musaelian[1,3], Simon Batzner[1,3] ✉, Anders Johansson[1], Lixin Sun[1], Cameron J. Owen[1], Mordechai Kornbluth[2] & Boris Kozinsky[1,2] ✉

A simultaneously accurate and computationally efficient parametrization of the potential energy surface of molecules and materials is a long-standing goal in the natural sciences. While atom-centered message passing neural networks (MPNNs) have shown remarkable accuracy, their information propagation has limited the accessible length-scales. Local methods, conversely, scale to large simulations but have suffered from inferior accuracy. This work introduces Allegro, a strictly local equivariant deep neural network interatomic potential architecture that simultaneously exhibits excellent accuracy and scalability. Allegro represents a many-body potential using iterated tensor products of learned equivariant representations without atom-centered message passing. Allegro obtains improvements over state-of-the-art methods on QM9 and revMD17. A single tensor product layer outperforms existing deep MPNNs and transformers on QM9. Furthermore, Allegro displays remarkable generalization to out-of-distribution data. Molecular simulations using Allegro recover structural and kinetic properties of an amorphous electrolyte in excellent agreement with ab-initio simulations. Finally, we demonstrate parallelization with a simulation of 100 million atoms.

Molecular dynamics (MD) and Monte-Carlo (MC) simulation methods are a core pillar of computational chemistry, materials science, and biology. Common to a diverse set of applications ranging from energy materials[1] to protein folding[2] is the requirement that predictions of the potential energy and atomic forces must be both accurate and computationally efficient to faithfully describe the evolution of complex systems over long timescales. While first-principles methods such as density functional theory (DFT), which explicitly treat the electrons of the system, provide an accurate and transferable description of the system, they exhibit poor scaling with system size and thus limit practical applications to small systems and short simulation times. Classical force fields based on simple functions of atomic coordinates are able to scale to large systems and long timescales but are inherently limited in their fidelity and can yield unfaithful dynamics. Descriptions of the potential energy surface (PES) using machine learning (ML) have emerged as a promising approach to move past this trade-off[3–24]. Machine learning interatomic potentials (MLIPs) aim to approximate a set of high-fidelity energy and force labels with improved computational efficiency that scales linearly in the number of atoms. A variety of approaches have been proposed, from shallow neural networks and kernel-based approaches[3–6] to more recent methods based on deep learning[14,15,20,25,26]. In particular, a class of MLIPs based on atom-centered message-passing neural networks (MPNNs) has shown remarkable accuracy[9,11,14,15,26,27]. In interatomic potentials based on MPNNs, an atomistic graph is induced by connecting each atom (node) to all neighboring atoms inside a finite cutoff sphere surrounding it. Information is then iteratively propagated along this graph, allowing MPNNs to learn many-body correlations and access non-local information outside of the local cutoff. This iterated propagation, however, leads to large receptive fields with many effective neighbors for each atom, which impedes parallel computation and limits the length scales accessible to atom-centered message-passing MLIPs. MLIPs using strictly local descriptors such as Behler-Parrinello neural networks[5], GAP[6], SNAP[7], DeepMD[20], Moment Tensor Potentials[8],

[1]Harvard University, Cambridge, MA, USA. [2]Robert Bosch LLC Research and Technology Center, Cambridge, MA, USA. [3]These authors contributed equally: Albert Musaelian, Simon Batzner. ✉e-mail: batzner@g.harvard.edu; bkoz@seas.harvard.edu

or ACE[12] do not suffer from this obstacle due to their strict locality. As a result, they can be easily parallelized across devices and have been successfully scaled to extremely large system sizes[28–31]. Approaches based on local descriptors, however, have so far fallen behind in accuracy compared to state-of-the-art equivariant, atom-centered message passing interatomic potentials[15].

Message-passing interatomic potentials

Message-passing neural networks (MPNNs) which learn atomistic representations have recently gained popularity in atomistic machine learning due to advantages in accuracy compared to hand-crafted descriptors. Atom-centered message-passing interatomic potentials operate on an atomistic graph constructed by representing atoms as nodes and defining edges between atoms that are within a fixed cutoff distance of one another. Each node is then represented by a hidden state $\mathbf{h}_i^t \in \mathbb{R}^c$ representing the state of atom i at layer t, and edges are represented by edge features $\mathbf{e}_{ij}$, for which the interatomic distance r_{ij} is often used. The message-passing formalism can then be concisely described as[32]:

$$\mathbf{m}_i^{t+1} = \sum_{j \in \mathcal{N}(i)} M_t\left(\mathbf{h}_i^t, \mathbf{h}_j^t, \mathbf{e}_{ij}\right) \tag{1}$$

$$\mathbf{h}_i^{t+1} = U_t\left(\mathbf{h}_i^t, \mathbf{m}_i^{t+1}\right) \tag{2}$$

where M_t and U_t are an arbitrary message function and node update function, respectively. From this propagation mechanism, it is immediately apparent that as messages are communicated over a sequence of t steps, the local receptive field of an atom i, i.e., the effective set of neighbors that contribute to the final state of atom i, grows approximately cubically with the effective cutoff radius $r_{c,e}$. In particular, given a MPNN with N_{layer} message-passing steps and local cutoff radius of $r_{c,l}$, the effective cutoff is $r_{c,e} = N_{\text{layer}} r_{c,l}$. Information from all atoms inside this receptive field feeds into a central atom's state $\mathbf{h}_i$ at the final layer of the network. Due to the cubic growth of the number of atoms inside the receptive field cutoff $r_{c,e}$, parallel computation can quickly become unmanageable, especially for extended periodic systems. As an illustrative example, we may take a structure of 64 molecules of liquid water at pressure $P = 1$ bar and temperature $T = 300$ K. For a typical setting of $N_t = 6$ message-passing layers with a local cutoff of $r_{c,l} = 6$ Å this would result in an effective cutoff of $r_{c,e} = 36$ Å. While each atom only has approximately 96 atoms in its local 6 Å environment (including the central atom), it has 20,834 atoms inside the extended 36 Å environment. Due to the atom-centered message-passing mechanism, information from each of these atoms flows into the current central atom. In a parallel scheme, each worker must have access to the high-dimensional feature vectors $\mathbf{h}_i$ of all 20,834 nodes, while the strictly local scheme only needs to have access to approximately $6^3 = 216$ times fewer atoms' states. From this simple example, it becomes obvious that massive improvements in scalability can be obtained from strict locality in machine learning interatomic potentials. It should be noted that conventional, atom-centered message passing allows for the possibility, in principle, to capture long-range interactions (up to $r_{c,e}$) and can induce many-body correlations. The relative importance of these effects in describing molecules and materials is an open question, and one of the aims of this work is to explore whether many-body interactions can be efficiently captured without increasing the effective cutoff.

Equivariant neural networks

The physics of atomic systems is unchanged under the action of a number of geometric symmetries—rotation, inversion, and translation—which together comprise the Euclidean group $E(3)$ (rotation alone is $SO(3)$, and rotation and inversion together comprise $O(3)$). Scalar quantities such as the potential energy are invariant to these symmetry group operations, while vector quantities such as the atomic forces are equivariant to them and transform correspondingly when the atomic geometry is transformed. More formally, a function between vector spaces $f: X \to Y$ is equivariant to a group G if

$$f(D_X[g]x) = D_Y[g]f(x) \quad \forall g \in G, \forall x \in X \tag{3}$$

where $D_X[g] \in GL(X)$ is the representation of the group element g in the vector space X. The function f is invariant if $D_Y[g]$ is the identity operator on Y: in this case, the output is unchanged by the action of symmetry operations on the input x.

Most existing MLIPs guarantee the invariance of their predicted energies by acting only on invariant inputs. In invariant, atom-centered message-passing interatomic potentials in particular, each atom's hidden latent space is a feature vector consisting solely of invariant scalars[25]. More recently, however, a class of models known as equivariant neural networks[33–36] have been developed which can act directly on non-invariant geometric inputs, such as displacement vectors, in a symmetry-respecting way. This is achieved by using only $E(3)$-equivariant operations, yielding a model whose internal features are equivariant with respect to the 3D Euclidean group. Building on these concepts, equivariant architectures have been explored for developing interatomic potential models. Notably, the NequIP model[15], followed by several other equivariant implementations[26,27,37–39], demonstrated unprecedentedly low error on a large range of molecular and materials systems, accurately describes structural and kinetic properties of complex materials, and exhibits remarkable sample efficiency. In both the present work and in NequIP, the representation $D_X[g]$ of an operation $g \in O(3)$ on an internal feature space X takes the form of a direct sum of irreducible representations (commonly referred to as irreps) of $O(3)$. This means that the feature vectors themselves are comprised of various geometric tensors corresponding to different irreps that describe how they transform under symmetry operations. The irreps of $O(3)$, and thus the features, are indexed by a rotation order $\ell \geq 0$ and a parity $p \in (-1, 1)$. A tensor that transforms according to the irrep ℓ, p is said to "inhabit" that irrep. We note that in many cases one may also omit the parity index to instead construct features that are only $SE(3)$-equivariant (translation and rotation), which simplifies the construction of the network and reduces the memory requirements.

A key operation in such equivariant networks is the tensor product of representations, an equivariant operation that combines two tensors $\mathbf{x}$ and $\mathbf{y}$ with irreps ℓ_1, p_1 and ℓ_2, p_2 to give an output inhabiting an irrep $\ell_{\text{out}}, p_{\text{out}}$ satisfying $|\ell_1 - \ell_2| \leq \ell_{\text{out}} \leq |\ell_1 + \ell_2|$ and $p_{\text{out}} = p_1 p_2$:

$$(\mathbf{x} \otimes \mathbf{y})_{\ell_{\text{out}}, m_{\text{out}}} = \sum_{m_1, m_2} \begin{pmatrix} \ell_1 & \ell_2 & \ell_{\text{out}} \\ m_1 & m_2 & m_{\text{out}} \end{pmatrix} \mathbf{x}_{\ell_1, m_1} \mathbf{y}_{\ell_2, m_2} \tag{4}$$

where $\begin{pmatrix} \ell_1 & \ell_2 & \ell_{\text{out}} \\ m_1 & m_2 & m_{\text{out}} \end{pmatrix}$ is the Wigner 3j symbol. Two key properties of the tensor product are that it is bilinear (linear in both $\mathbf{x}$ and $\mathbf{y}$) and that it combines tensors inhabiting different irreps in a symmetrically valid way. Many simple operations are encompassed by the tensor product, such as for example:

- scalar-scalar multiplication: $(\ell_1 = 0, p_1 = 1), (\ell_2 = 0, p_2 = 1) \to (\ell_{\text{out}} = 0, p_{\text{out}} = 1)$
- vector dot product: $(\ell_1 = 1, p_1 = -1), (\ell_2 = 1, p_2 = -1) \to (\ell_{\text{out}} = 0, p_{\text{out}} = 1)$
- vector cross product, resulting in a pseudovector: $(\ell_1 = 1, p_1 = -1), (\ell_2 = 1, p_2 = -1) \to (\ell_{\text{out}} = 1, p_{\text{out}} = 1)$.

The message function $M_t(\mathbf{h}_i^t, \mathbf{h}_j^t, \mathbf{e}_{ij})$ of the NequIP model, for example, uses this tensor product to define a message from atom j to i

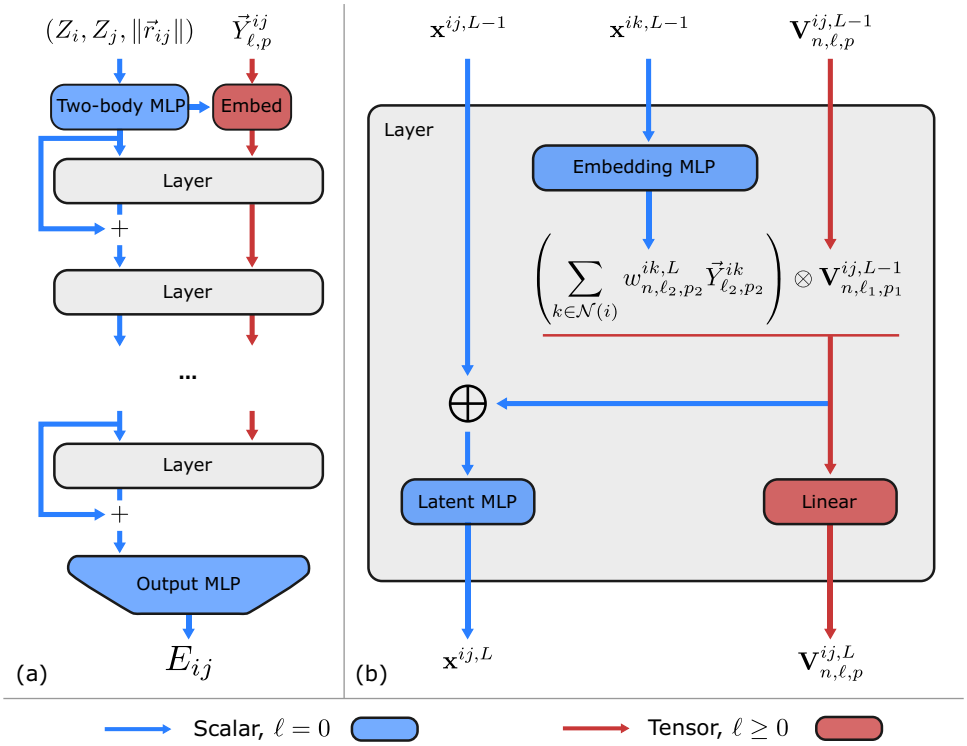

Fig. 1 | The Allegro network. a shows the Allegro model architecture and **b** details a tensor product layer. Blue and red arrows represent scalar and tensor information, respectively, ⊗ denotes the tensor product, and ⊕ is concatenation.

as a tensor product between equivariant features of the edge *ij* and the equivariant features of the neighboring node *j*.

Atomic cluster expansion

Finally, parallel to atom-centered message-passing interatomic potentials, the Atomic Cluster Expansion (ACE) has been developed as a unifying framework for various descriptor-based MLIPs[12]. ACE can also be expressed in terms of the same tensor product operation introduced above, with further details provided in "Methods".

In this work, we present Allegro, an equivariant deep-learning approach that retains the high accuracy of the recently proposed class of equivariant MPNNs[15,26,27,37,39,40] while combining it with strict locality and thus the ability to scale to large systems. We demonstrate that Allegro not only obtains state-of-the-art accuracy on a series of different benchmarks but can also be parallelized across devices to access simulations with hundreds of millions of atoms. We further find that Allegro displays a high level of transferability to out-of-distribution data, significantly outperforming other local MLIPs, in particular including body-ordered approaches. Finally, we show that Allegro can faithfully recover structural and kinetic properties from molecular dynamics simulations of Li_3PO_4, a complex phosphate electrolyte.

The outline of the article is as follows: we first surveyed relevant related work on message-passing interatomic potentials, equivariant neural networks, and the atomic cluster expansion. We then outline the core ideas and design of the Allegro approach, followed by a series of results on standard benchmarks. Finally, we show the results of molecular dynamics simulations on a challenging material, an analysis of the scaling properties of Allegro, and a theoretical analysis of the framework.

Results

In the following, we describe the proposed method for learning high-dimensional potential energy surfaces using strictly local many-body equivariant representations.

Energy decomposition

We start by decomposing the potential energy of a system into per-atom energies E_i, in line with previous approaches[5,6,25]:

$$E_{\text{system}} = \sum_i^N \sigma_{Z_i} E_i + \mu_{Z_i} \tag{5}$$

where σ_{Z_i} and μ_{Z_i} are per-species scale and shift parameters, which may be trainable. Unlike most existing MLIPs, we further decompose the per-atom energy into a sum of pairwise energies, indexed by the central atom and one of its local neighbors

$$E_i = \sum_{j\in\mathcal{N}(i)} \sigma_{Z_i,Z_j} E_{ij} \tag{6}$$

where j ranges over the neighbors of atom i, and again one may optionally apply a per-species-pair scaling factor σ_{Z_i,Z_j}. It is important to note that while these pairwise energies are indexed by the atom i and its neighbor j, they may depend on all neighboring atoms k belonging to the local environment $\mathcal{N}(i)$. Because E_{ij} and E_{ji} contribute to different site energies E_i and E_j, respectively, we choose that they depend only on the environments of the corresponding central atoms. As a result and by design, $E_{ij} \neq E_{ji}$. Finally, the force acting on atom i, $\vec{F}_i$, is computed using autodifferentiation according to its definition as the negative gradient of the total energy with respect to the position of atom i:

$$\vec{F}_i = -\nabla_i E_{\text{system}}$$

which gives an energy-conserving force field.

The Allegro model

The Allegro architecture, shown in Fig. 1, is an arbitrarily deep equivariant neural network with $N_{\text{layer}} \geq 1$ layers. The architecture learns representations associated with ordered pairs of neighboring atoms

using two latent spaces: an invariant latent space, which consists of scalar ($\ell = 0$) features, and an equivariant latent space, which processes tensors of arbitrary rank $\ell \geq 0$. The two latent spaces interact with each other at every layer. The final pair energy E_{ij} is then computed by a multi-layer perceptron (MLP) acting on the final layer's scalar features.

We use the following notations:

$\vec{r}_i$: position of the ith atom in the system

$\vec{r}_{ij}$: relative displacement vector $\vec{r}_j - \vec{r}_i$ from i to j

r_{ij}: corresponding interatomic distance

$\hat{r}_{ij}$: unit vector of $\vec{r}_{ij}$

$Y_{\ell,p}$: projection of $\hat{r}_{ij}$ onto the ℓ-th real spherical harmonic which has parity $p = (-1)^\ell$. We omit the $m = -\ell, \ldots, 0, \ldots \ell$ index within the representation from the notation for compactness

Z_i: chemical species of atom i

MLP(...): multi-layer perceptron—a fully connected scalar neural network, possibly with nonlinearities

$\mathbf{x}^{ij,L}$: invariant scalar latent features of the ordered pair of atoms ij at layer L

$\mathbf{V}^{ij,L}_{n,\ell,p}$: equivariant latent features of the ordered pair of atoms ij at layer L. These transform according to a direct sum of irreps indexed by the rotation order $\ell \in 0, 1, \ldots, \ell_{\max}$ and parity $p \in -1, 1$ and thus consist of both scalars ($\ell = 0$) and higher-order tensors ($\ell > 0$). The hyperparameter $\ell_{\max}$ controls the maximum rotation order to which features in the network are truncated. In Allegro, n denotes the channel index which runs over $0, \ldots, n_{\text{equivariant}} - 1$. We omit the m index within each irreducible representation from the notation for compactness.

Two-body latent embedding. Before the first tensor product layer, the scalar properties of the pair ij are embedded through a nonlinear MLP to give the initial scalar latent features $\mathbf{x}^{ij,L=0}$:

$$\mathbf{x}^{ij,L=0} = \text{MLP}_{\text{two-body}}\left(\text{1Hot}(Z_i) \parallel \text{1Hot}(Z_j) \parallel B(r_{ij})\right) \cdot u(r_{ij}) \tag{7}$$

where $\parallel$ denotes concatenation, $\text{1Hot}(\cdot)$ is a one-hot encoding of the center and neighbor atom species Z_i and Z_j, and

$$B(r_{ij}) = (B_1(r_{ij}) \parallel \ldots \parallel B_{N_{\text{basis}}}(r_{ij}))) \tag{8}$$

is the projection of the interatomic distance r_{ij} onto a radial basis. We use the Bessel basis functions with a polynomial envelope function as proposed in ref. [14]. The basis is normalized as described in Supplementary Note 1 and plotted in Supplementary Fig. 1. Finally, the function $u(r_{ij}) : \mathbb{R} \to \mathbb{R}$ by which the output of $\text{MLP}_{\text{two-body}}$ is multiplied is the same smooth cutoff envelope function as used in the radial basis function.

The initial equivariant features $\mathbf{V}^{ij,L=0}_{n,\ell,p}$ are computed as a linear embedding of the spherical harmonic projection of $\hat{r}_{ij}$:

$$\mathbf{V}^{ij,L=0}_{n,\ell,p} = w^{ij,L=0}_{n,\ell,p} \vec{Y}^{ij}_{\ell,p} \tag{9}$$

where the channel index is $n = 0, \ldots, n_{\text{equivariant}} - 1$, and where the scalar weights $w^{ij,L=0}_{n,\ell,p}$ for each center-neighbor pair ij are computed from the initial two-body scalar latent features:

$$w^{ij,L=0}_{n,\ell,p} = \text{MLP}^{L=0}_{\text{embed}}(\mathbf{x}^{ij,L=0})_{n,\ell,p}. \tag{10}$$

Layer architecture. Each Allegro tensor product layer consists of four components:

1. an MLP that generates weights to embed the central atom's environment
2. an equivariant tensor product using those weights
3. an MLP to update the scalar latent space with scalar information resulting from the tensor product
4. an equivariant linear layer that mixes channels in the equivariant latent space.

Tensor product: Our goal is to incorporate interactions between the current equivariant state of the center-neighbor pair and other neighbors in the environment, and the most natural operation with which to interact equivariant features is the tensor product. We thus define the updated equivariant features on the pair ij as a weighted sum of the tensor products of the current features with the geometry of the various other neighbor pairs ik in the local environment of i:

$$\mathbf{V}^{ij,L}_{n,(\ell_1,p_1,\ell_2,p_2)\to(\ell_{\text{out}},p_{\text{out}})} = \sum_{k\in\mathcal{N}(i)} w^{ik,L}_{n,\ell_2,p_2}\left(\mathbf{V}^{ij,L-1}_{n,\ell_1,p_1} \otimes \vec{Y}^{ik}_{\ell_2,p_2}\right) \tag{11}$$

$$= \sum_{k\in\mathcal{N}(i)} \mathbf{V}^{ij,L-1}_{n,\ell_1,p_1} \otimes \left(w^{ik,L}_{n,\ell_2,p_2} \vec{Y}^{ik}_{\ell_2,p_2}\right) \tag{12}$$

$$= \mathbf{V}^{ij,L-1}_{n,\ell_1,p_1} \otimes \left(\sum_{k\in\mathcal{N}(i)} w^{ik,L}_{n,\ell_2,p_2} \vec{Y}^{ik}_{\ell_2,p_2}\right) \tag{13}$$

In the second and third lines, we exploit the bilinearity of the tensor product in order to express the update in terms of one tensor product, rather than one for each neighbor k, which saves significant computational effort. This is a variation on the "density trick"[6,41].

We note that valid tensor product paths are all those satisfying $|\ell_1 - \ell_2| \leq \ell_{\text{out}} \leq |\ell_1 + \ell_2|$ and $p_{\text{out}} = p_1 p_2$, so it is possible to have $(\ell_1, p_1) \neq (\ell_2, p_2) \neq (\ell_{\text{out}}, p_{\text{out}})$. We additionally enforce $\ell_{\text{out}} \leq \ell_{\max}$. Which tensor product paths to include is a hyperparameter choice. In this work we include all allowable paths but other choices, such as restricting $(\ell_{\text{out}}, p_{\text{out}})$ to be among the values of (ℓ_1, p_1), are possible.

Environment embedding: The second argument to the tensor product, $\sum_{k\in\mathcal{N}(i)} w^{ik,L}_{n,\ell_2,p_2} \vec{Y}^{ik}_{\ell_2,p_2}$, is a weighted sum of the spherical harmonic projections of the various neighbor atoms in the local environment. This can be viewed as a weighted spherical harmonic basis projection of the atomic density, similar to the projection onto a spherical-radial basis used in ACE[12] and SOAP[41]. For this reason, we refer to $\sum_{k\in\mathcal{N}(i)} w^{ik,L}_{n,\ell_2,p_2} \vec{Y}^{ik}_{\ell_2,p_2}$ as the "embedded environment" of atom i.

A central difference from the atomic density projections used in descriptor methods, however, is that the weights of the sum are learned. In descriptor approaches such as ACE, the n index runs over a pre-determined set of radial–chemical basis functions, which means that the size of the basis must increase with both the number of species and the desired radial resolution. In Allegro, we instead leverage the previously learned scalar featurization of each center-neighbor pair to further learn

$$w^{ik,L}_{n,\ell_2,p_2} = \text{MLP}^{L}_{\text{embed}}(\mathbf{x}^{ik,L-1})_{n,\ell_2,p_2} \tag{14}$$

which yields an embedded environment with a fixed, chosen number of channels $n_{\text{equivariant}}$. It is important to note that $w^{ik,L}_{n,\ell_2,p_2}$ is learned as a function of the existing scalar latent representation of the center-neighbor pair ik from previous layers. At later layers, this can contain significantly more information about the environment of i than a two-body radial basis. We typically choose $\text{MLP}_{\text{embed}}$ to be a simple one-layer linear projection of the scalar latent space.

Latent MLP: Following the tensor product defined in Eq. (11), the scalar outputs of the tensor product are reintroduced into the scalar

Table 1 | MAE on the revised MD-17 dataset for energies and force components, in units of [meV] and [meV/Å], respectively

Molecule		FCHL19[13, 43]	UNiTE[26]	GAP[6]	ANI-pretrained[48, 49]	ANI-random[48, 49]	ACE[12]	GemNet-(T/Q)[76]	NequIP (l=3)[15]	Allegro
Aspirin	Energy	6.2	2.4	17.7	16.6	25.4	6.1	–	**2.3**	**2.3**
	Forces	20.9	7.6	44.9	40.6	75.0	17.9	9.5	8.2	**7.3**
Azobenzene	Energy	2.8	1.1	8.5	15.9	19.0	3.6	–	**0.7**	1.2
	Forces	10.8	4.2	24.5	35.4	52.1	10.9	–	2.9	**2.6**
Benzene	Energy	0.3	0.07	0.75	3.3	3.4	**0.04**	–	**0.04**	0.3
	Forces	2.6	0.73	6.0	10.0	17.5	0.5	0.5	0.3	**0.2**
Ethanol	Energy	0.9	0.62	3.5	2.5	7.7	1.2	–	**0.4**	**0.4**
	Forces	6.2	3.7	18.1	13.4	45.6	7.3	3.6	2.8	**2.1**
Malonaldehyde	Energy	1.5	1.1	4.8	4.6	9.4	1.7	–	0.8	**0.6**
	Forces	10.2	6.6	26.4	24.5	52.4	11.1	6.6	5.1	**3.6**
Naphthalene	Energy	1.2	0.46	3.8	11.3	16.0	0.9	–	**0.2**	0.5
	Forces	6.5	2.6	16.5	29.2	52.2	5.1	1.9	1.3	**0.9**
Paracetamol	Energy	2.9	1.9	8.5	11.5	18.2	4.0	–	**1.4**	1.5
	Forces	12.2	7.1	28.9	30.4	63.3	12.7	–	5.9	**4.9**
Salicylic acid	Energy	1.8	0.73	5.6	9.2	13.5	1.8	–	**0.7**	0.9
	Forces	9.5	3.8	24.7	29.7	52.0	9.3	5.3	4.0	**2.9**
Toluene	Energy	1.6	0.45	4.0	7.7	12.6	1.1	–	**0.3**	0.4
	Forces	8.8	2.5	17.8	24.3	52.9	6.5	2.2	**1.6**	1.8
Uracil	Energy	0.6	0.58	3.0	5.1	8.3	1.1	–	**0.4**	0.6
	Forces	4.2	3.8	17.6	21.4	44.1	6.6	3.8	3.1	**1.8**

Results for GAP, ANI, and ACE as reported in ref. [24]. Best results are marked in bold. ANI-pretrained refers to a version of ANI that was pretrained on 8.9 million structures and fine-tuned on the revMD-17 dataset, ANI-random refers to a randomly initialized model trained from scratch.

latent space as follows:

$$\mathbf{x}^{ij,L} = \mathrm{MLP}^{L}_{\mathrm{latent}}\left(\mathbf{x}^{ij,L-1} \,\|\, \bigoplus_{(\ell_1,p_1,\ell_2,p_2)} \mathbf{V}^{ij,L}_{n,(\ell_1,p_1,\ell_2,p_2)\to(\ell_{\mathrm{out}}=0,p_{\mathrm{out}}=1)}\right)\cdot u(r_{ij}) \tag{15}$$

where $\|$ denotes concatenation and $\oplus$ denotes concatenation over all tensor product paths whose outputs are scalars ($\ell_{\mathrm{out}}=0, p_{\mathrm{out}}=1$), each of which contributes $n_{\mathrm{equivariant}}$ scalars. The function $u(r_{ij}):\mathbb{R}\to\mathbb{R}$ is again the smooth cutoff envelope from Eq. (7). The purpose of the latent MLP is to compress and integrate information from the tensor product, whatever its dimension, into the fixed dimension invariant latent space. This operation completes the coupling of the scalar and equivariant latent spaces since the scalars taken from the tensor product contain information about non-scalars previously only available to the equivariant latent space.

Mixing equivariant features: Finally, the outputs of various tensor product paths with the same irrep ($\ell_{\mathrm{out}}, p_{\mathrm{out}}$) are linearly mixed to generate output equivariant features $\mathbf{V}^{ij,L}_{n,\ell,p}$ with the same number of channels indexed by n as the input features had:

$$\mathbf{V}^{ij,L}_{n,\ell,p} = \sum_{\substack{n'\\(\ell_1,p_1,\ell_2,p_2)}} w^{L}_{n,n',(\ell_1,p_1,\ell_2,p_2)\to(\ell,p)}\,\mathbf{V}^{ij,L}_{n',(\ell_1,p_1,\ell_2,p_2)\to(\ell,p)}. \tag{16}$$

The weights $w^{L}_{n,n',(\ell_1,p_1,\ell_2,p_2)\to(\ell,p)}$ are learned. This operation compresses the equivariant information from various paths with the same output irrep (ℓ,p) into a single output space regardless of the number of paths.

We finally note that an $SE(3)$-equivariant version of Allegro, which is sometimes useful for computational efficiency, can be constructed identically to the $E(3)$-equivariant model described here by simply omitting all parity subscripts p.

Residual update. After each layer, Allegro uses a residual update[42] in the scalar latent space that updates the previous scalar features from layer $L-1$ by adding the new features to them (see Supplementary Note 2). The residual update allows the network to easily propagate scalar information from earlier layers forward.

Output block. To predict the pair energy E_{ij}, we apply a fully connected neural network with output dimension 1 to the latent features output by the final layer:

$$E_{ij} = \mathrm{MLP}_{\mathrm{output}}(\mathbf{x}^{ij,L=N_{\mathrm{layer}}}) \tag{17}$$

Finally, we note that we found normalization, both of the targets and inside the network, to be of high importance. Details are outlined in "Methods".

Dynamics of small molecules

We benchmark Allegro's ability to accurately learn energies and forces of small molecules on the revised MD-17 dataset[43], a recomputed version of the original MD-17 dataset[10,44,45] that contains ten small, organic molecules at DFT accuracy. As shown in Table 1, Allegro obtains state-of-the-art accuracy in the mean absolute error (MAE) in force components, while NequIP performs better for the energies of some molecules. We note that while an older version of the MD-17 dataset which has widely been used to benchmark MLIPs exists[10,44,45], it has been observed to contain noisy labels[43] and is thus only of limited use for comparing the accuracy of MLIPs.

Transferability to higher temperatures

For an interatomic potential to be useful in practice, it is crucial that it be transferable to new configurations that might be visited over the course of a long molecular dynamics simulation. To assess Allegro's generalization capabilities, we test the transferability to conformations sampled from higher-temperature MD simulations. We use the temperature transferability benchmark introduced in ref. [24]: here, a series

Table 2 | Energy and Force RMSE for the 3BPA temperature transferability dataset, reported in units of [meV] and [meV/Å]

	ACE[12]	sGDML[10]	GAP[6]	FF[46, 47]	ANI-pretrained[48, 49]	ANI-2x[48, 49]	NequIP[15]	Allegro
Fit to 300 K								
300 K, E	7.1	9.1	22.8	60.8	23.5	38.6	**3.28 (0.12)**	3.84 (0.10)
300 K, F	27.1	46.2	87.3	302.8	42.8	84.4	**10.77 (0.28)**	12.98 (0.20)
600 K, E	24.0	484.8	61.4	136.8	37.8	54.5	**11.16 (0.17)**	12.07 (0.55)
600 K, F	64.3	439.2	151.9	407.9	71.7	102.8	**26.37 (0.11)**	29.11 (0.27)
1200 K, E	85.3	774.5	166.8	325.5	76.8	88.8	**38.52 (2.00)**	42.57 (1.79)
1200 K, F	187.0	711.1	305.5	670.9	129.6	139.6	**76.18 (1.36)**	82.96 (2.17)

All models were trained on $T = 300$ K. Results for all models except for NequIP and Allegro from ref. [24]. Best results are marked in bold. For NequIP and Allegro, we report the mean over three different seeds as well as the sample standard deviation in parentheses.

Table 3 | Comparison of models on the QM9 dataset, measured by the MAE in units of [meV]

Model	U_0	U	H	G
Schnet[25]	14	19	14	14
DimeNet++[77]	6.3	6.3	6.5	7.6
Cormorant[23]	22	21	21	20
LieConv[78]	19	19	24	22
L1Net[79]	13.5	13.8	14.4	14.0
SphereNet[80]	6.3	7.3	6.4	8.0
EGNN[40]	11	12	12	12
ET[38]	6.2	6.3	6.5	7.6
NoisyNodes[81]	7.3	7.6	7.4	8.3
PaiNN[27]	5.9	5.7	6.0	7.4
Allegro, 1 layer	5.7 (0.3)	5.3	5.3	6.6
Allegro, 3 layers	**4.7** (0.2)	**4.4**	**4.4**	**5.7**

Allegro outperforms all existing atom-centered message-passing and transformer-based models, in particular even with a single layer. Best methods are shown in bold.

of data were computed using DFT for the flexible drug-like molecule 3-(benzyloxy)pyridin-2-amine (3BPA) at temperatures 300, 600, and 1200 K. Various state-of-the-art methods were trained on 500 structures from the $T = 300$ K dataset and then evaluated on data sampled at all three temperatures. Table 2 shows a comparison of Allegro against existing approaches reported in ref. [24]: linear ACE[12], sGDML[10], GAP[6], a classical force field based on the GAFF functional form[46,47] as well as two ANI parametrizations[48,49] (ANI-pretrained refers to a version of ANI that was pretrained on 8.9 million structures and fine-tuned on this dataset, while ANI-2x refers to the original parametrization trained on 8.9 million structures, but not fine-tuned on the 3BPA dataset). The equivariant neural networks Allegro and NequIP are observed to generalize significantly better than all other approaches.

Quantum-chemical properties of small molecules

Next, we assess Allegro's ability to accurately model properties of small molecules across chemical space using the popular QM9 dataset[50]. The QM9 dataset contains molecular properties computed with DFT of approximately 134k minimum-energy structures with chemical elements (C, H, O, N, F) that contain up to 9 heavy atoms (C, O, N, F). We benchmark Allegro on four energy-related targets, in particular: (a) U_0, the internal energy of the system at $T = 0$ K, (b) U, the internal energy at $T = 298.15$ K, (c) H, the enthalpy at $T = 298.15$ K, and (d) G, the free energy at $T = 298.15$ K. Unlike other experiments in this work, which probed conformational degrees of freedom, we here assess the ability of Allegro to describe properties across compositional degrees of freedom. Table 3 shows a comparison with a series of state-of-the-art methods that also learn the properties described above as a direct mapping from atomic coordinates and species. We find that Allegro outperforms all existing methods. Surprisingly, even an Allegro model with a single tensor product layer obtains higher accuracy than all existing methods based on atom-centered message-passing neural networks and transformers.

Li-ion diffusion in a phosphate electrolyte

In order to examine Allegro's ability to describe kinetic properties with MD simulations, we use it to study amorphous structure formation and Li-ion migration in the Li_3PO_4 solid electrolyte. This class of solid-state electrolytes is characterized by the intricate dependence of conductivity on the degree of crystallinity[51–54].

In particular, the Li_3PO_4 dataset used in this work consists of two parts: a 50 ps ab-initio molecular dynamics (AIMD) simulation in the molten liquid state at $T = 3000$ K, followed by a 50 ps AIMD simulation in the quenched state at $T = 600$ K. We train a potential on structures from the liquid and quenched trajectories. The model used here is computationally efficient due to a relatively small number of parameters (9058 weights) and tensor products. In particular, we note that the model used to measure the faithfulness of the kinetics and to measure Allegro's ability to predict thermodynamic observables is identical to the one used in scaling experiments detailed below. This is crucial for fair assessment of a method that simultaneously scales well and can accurately predict material properties. When evaluated on the test set for the quenched amorphous state, which the simulation is performed on, a MAE in the energies of 1.7 meV/atom was obtained, as well as a MAE in the force components of 73.4 meV/Å. We then run a series of ten MD simulations starting from the initial structure of the quenched AIMD simulation, all of length 50 ps at $T = 600$ K in the quenched state, in order to examine how well Allegro recovers the structure and kinetics compared to AIMD. To assess the quality of the structure after the phase change, we compare the all-atom radial distribution functions (RDF) and the angular distribution functions (ADF) of the tetrahedral angle P–O–O (P central atom). We show in Fig. 2 that Allegro can accurately recover both distribution functions. For the aspect of ion transport kinetics, we test how well Allegro can model the Li mean-square-displacement (MSD) in the quenched state. We again find excellent agreement with AIMD, as shown in Fig. 3. The structure of Li_3PO_4 can be seen Fig. 4.

Scaling

Many interesting phenomena in materials science, chemistry, and biology require large numbers of atoms, long timescales, a diversity of chemical elements, or often all three. Scaling to large numbers of atoms requires parallelization across multiple workers, which is difficult in atom-centered MPNNs because the iterative propagation of atomic state information along the atomistic graph increases the size of the receptive field as a function of the number of layers. This is further complicated by the fact that access to energy-conservative

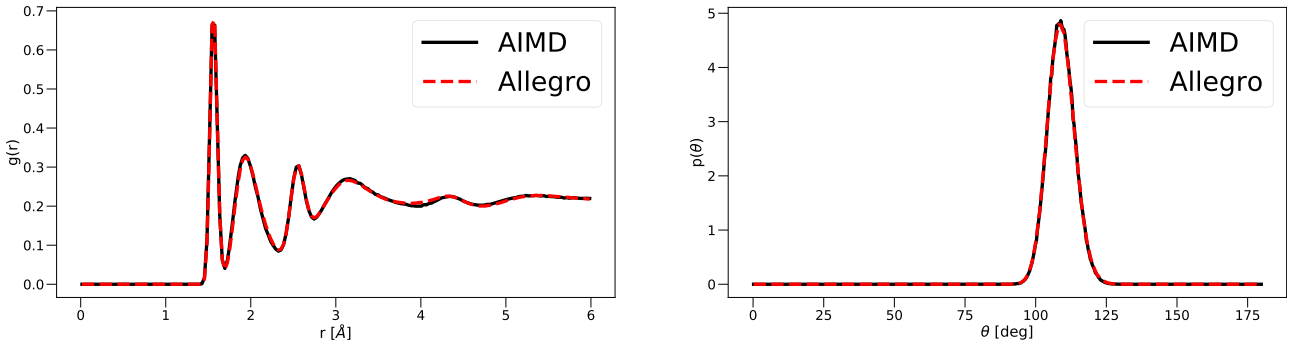

Fig. 2 | Structural properties of Li_3PO_4. Left: radial distribution function, right: angular distribution function of tetrahedral bond angle. All defined as probability density functions. Results from Allegro are shown in red, and those from AIMD are shown in black.

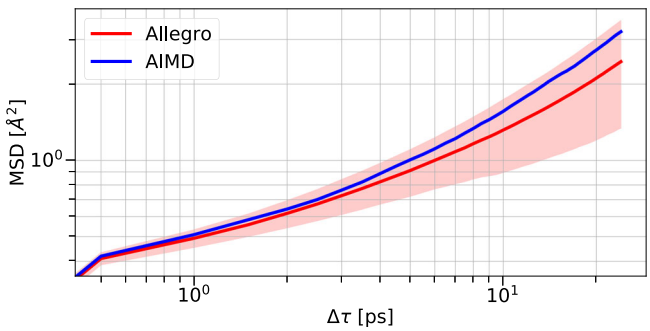

Fig. 3 | Li dynamics in Li_3PO_4. Comparison of the Li MSD of AIMD vs. Allegro. Results are averaged over 10 runs of Allegro, shading indicates +/− one standard deviation. Results from Allegro are shown in red, and those from AIMD are shown in blue.

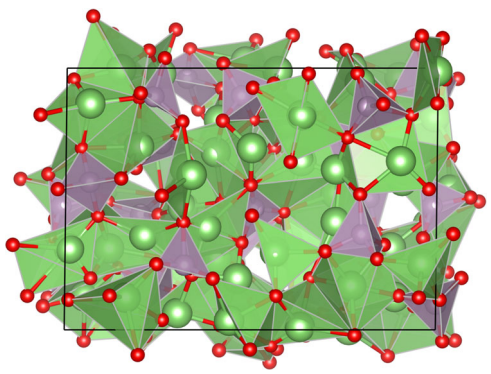

Fig. 4 | Structure of Li_3PO_4. The quenched Li_3PO_4 structure at $T = 600$ K.

force fields requires computing the negative gradient of the predicted energy, which in standard backpropagation algorithms also requires propagating gradient information along the atom graph. Allegro is designed to avoid this issue by strict locality. A given Allegro model scales as:

- $\mathcal{O}(N)$ in the number of atoms in the system N, in contrast to the $\mathcal{O}(N^2)$ scaling of some global descriptor methods such as sGDML[10];
- $\mathcal{O}(M)$ in the number of neighbors per atom M, in contrast to the quadratic $\mathcal{O}(M^2)$ scaling of some deep-learning approaches such as DimeNet[14] or Equivariant Transformers[38,55];
- $\mathcal{O}(1)$ in the number of species S, unlike local descriptors such as SOAP ($\mathcal{O}(S^2)$) or ACE ($\mathcal{O}(S^{\text{bodyorder}-1})$)[12].

We note, however, that the per-pair featurization of Allegro has larger memory requirements than if one were to choose the same number of features in a per-atom featurization. In practice, we find this to not be a problem and see that Allegro can be scaled to massive systems by parallelizing over modest computational resources.

In particular, in addition to scaling as $\mathcal{O}(N)$ in the number of atoms, Allegro is strictly local within the chosen cutoff and thus easy to parallelize in large-scale calculations. Recall that Eqs. (5) and (6) define the total energy of a system in Allegro as a sum over scaled pairwise energies E_{ij}. Thus by linearity, the force on atom a

$$\vec{F}_a = -\nabla_a E_{\text{system}} = -\sum_{ij} \nabla_a E_{ij},$$

ignoring the per-species and per-species-pair scaling coefficients σ_{Z_i} and σ_{Z_i,Z_j} for clarity. Because each E_{ij} depends only the atoms in the neighborhood of atom i, $-\nabla_a E_{ij} \neq 0$ only when a is in the neighborhood of i. Further, for the same reason, pair energy terms E_{ij} with different central atom indices i are independent. As a result, these groups of terms may be computed independently for each central atom, which facilitates parallelization: the contributions to the force on atom a due to the neighborhoods of various different atoms can be computed in parallel by whichever worker is currently assigned the relevant center's neighborhood. The final forces are then simple sum reductions over force terms from various parallel workers.

We first demonstrate the favorable scaling of Allegro in system size by parallelizing the method across GPUs on a single compute node as well as across multiple GPU nodes. We choose two test systems for the scaling experiments: (a) the quenched state structures of the multi-component electrolyte Li_3PO_4 and (b) the Ag bulk crystal with a vacancy, simulated at 90% of the melting temperature. The Ag model used 1000 structures for training and validation, resulting in energy MAE of 0.397 meV/atom and force MAE of 16.8 meV/Å on a test set of 159 structures. Scaling numbers are dependent on a variety of hyper-parameter choices, such as network size and radial cutoff, that control the trade-off between evaluation speed and accuracy. For Li_3PO_4, we explicitly choose these identically to those used in the previous set of experiments in order to demonstrate how well an Allegro potential scales that we demonstrated to give highly accurate prediction of structure and kinetics. Table 4 shows the computational efficiency for varied size and computational resources. We are able to simulate the Ag system with over 100 million atoms on 16 GPU nodes.

The parallel nature of the method and its implementation also allows multiple GPUs to be used to increase the speed of the potential calculation for a fixed-size system. Figure 5 shows such strong scaling

Table 4 | Simulation times obtainable in [ns/day] and time required per atom per step in [microseconds] for varying number of atoms and computational resources

Material	Number of atoms	Number of GPUs	Speed in ns/day	Microseconds/(atom · step)
Li_3PO_4	192	1	32.391	27.785
Li_3PO_4	421,824	1	0.518	0.552
Li_3PO_4	421,824	2	1.006	0.284
Li_3PO_4	421,824	4	1.994	0.143
Li_3PO_4	421,824	8	3.810	0.075
Li_3PO_4	421,824	16	6.974	0.041
Li_3PO_4	421,824	32	11.530	0.025
Li_3PO_4	421,824	64	15.515	0.018
Li_3PO_4	50,331,648	128	0.274	0.013
Ag	71	1	90.190	67.463
Ag	1,022,400	1	1.461	0.289
Ag	1,022,400	2	2.648	0.160
Ag	1,022,400	4	5.319	0.079
Ag	1,022,400	8	10.180	0.042
Ag	1,022,400	16	18.812	0.022
Ag	1,022,400	32	28.156	0.015
Ag	1,022,400	64	43.438	0.010
Ag	1,022,400	128	49.395	0.009
Ag	100,640,512	128	1.539	0.003

Time steps of 2fs and 5fs were used for Li_3PO_4 and Ag, respectively.

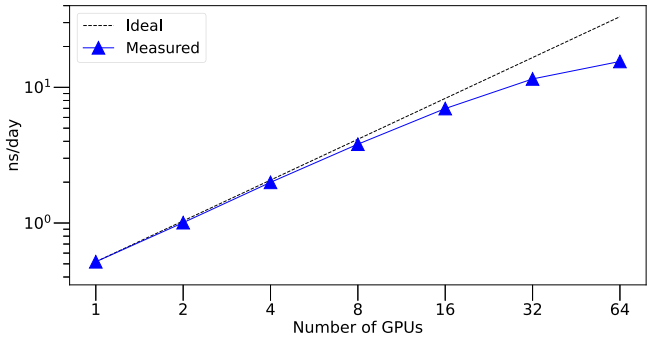

Fig. 5 | Scaling results. Strong scaling results on a Li_3PO_4 structure of 421,824 atoms, performed in LAMMPS.

results on a 421,824 atom Li_3PO_4 structure. The system size was kept constant while varying the number of A100 GPUs.

Theoretical analysis

In this section, we provide a theoretical analysis of the method by highlighting similarities and differences to the Atomic Cluster Expansion (ACE) framework[12]. Throughout this section we omit representation indices ℓ and p from the notation for conciseness: every weight or feature that carries ℓ and p indices previously implicitly carries them in this section. Starting from the initial equivariant features for the pair of atoms ij at layer $L = 0$

$$\mathbf{V}_{n_0}^{ij,L=0} = w_{n_0}^{ij,L=0} \vec{Y}^{ij} \tag{18}$$

the first Allegro layer computes a sum over tensor products between $\mathbf{V}_{n_0}^{ij,L=0}$ and the spherical harmonics projection of all neighbors $k \in \mathcal{N}(i)$:

$$\mathbf{V}_{n_1}^{ij,L=1} = \sum_{\substack{n_1' \\ \text{paths}}} w_{n_1,n_1',\text{path}}^{L=1} \sum_{k_1 \in \mathcal{N}(i)} w_{n_1'}^{ik_1,L=1} \left(w_{n_1'}^{ij,L=0} \vec{Y}^{ij} \otimes \vec{Y}^{ik_1} \right) \tag{19}$$

$$= \sum_{\substack{n_1' \\ \text{paths}}} w_{n_1,n_1',\text{path}}^{L=1} \sum_{k_1 \in \mathcal{N}(i)} w_{n_1'}^{ik_1,L=1} w_{n_1'}^{ij,L=0} \left(\vec{Y}^{ij} \otimes \vec{Y}^{ik_1} \right) \tag{20}$$

which follows from the bilinearity of the tensor product. The sum over "paths" in this equation indicates the sum over all symmetrically valid combinations of implicit irrep indices on the various tensors present in the equation as written out explicitly in Eq. (16). Repeating this substitution, we can express the equivariant features at layer $L = 2$ and reveal a general recursive relationship:

$$\mathbf{V}_{n_2,\ell_2,p_2}^{ij,L=2} = \sum_{\substack{n_2' \\ \text{paths}}} w_{n_2,n_2',\text{path}}^{L=2} \sum_{k_2 \in \mathcal{N}(i)} w_{n_2'}^{ik_2,L=2} \left(\mathbf{V}_{n_2'}^{ij,L=1} \otimes \vec{Y}^{ik_2} \right) \tag{21}$$

$$= \sum_{\substack{n_1',n_2' \\ \text{paths}}} w_{n_2,n_2',\text{path}}^{L=2} w_{n_2',n_1',\text{path}}^{L=1} \left[\sum_{k_2 \in \mathcal{N}(i)} \sum_{k_1 \in \mathcal{N}(i)} w_{n_2'}^{ik_2,L=2} w_{n_1'}^{ik_1,L=1} w_{n_1'}^{ij,L=0} \left(\vec{Y}^{ij} \otimes \vec{Y}^{ik_1} \otimes \vec{Y}^{ik_2} \right) \right] \tag{22}$$

$$\mathbf{V}_{n_L,\ell_L,p_L}^{ij,L} = \sum_{\substack{k_1,\ldots,k_L \\ n_1',\ldots,n_L' \\ \text{paths}}} \left[\left(\prod_{\alpha \in 1,\ldots,L} w_{n_{\alpha+1}',n_\alpha',\text{path}}^{L=\alpha} \right) \left(\prod_{\alpha \in 0,\ldots,L} w_{n_\alpha'}^{ik_\alpha,L=\alpha} \right) \left(\bigotimes_{\alpha \in 0,\ldots,L} \vec{Y}^{ik_\alpha} \right) \right] \tag{23}$$

where $k_0 = j$, $n_{L+1}' = n_L$, and $n_0' = n_1'$.

The ACE descriptor $B_{n_1\ldots n_\nu}^{(\nu)}$ of body order $\nu + 1$[12] can also be written as an iterated tensor product, specifically of the projection A_n of the local atomic density onto a spherical harmonic and radial–chemical basis. The n index here runs over the $N_{\text{full-basis}} = S \times N_{\text{basis}}$ combined radial–chemical basis functions. Starting from this definition we may again use the bilinearity of the tensor product to expand the ACE descriptor:

$$B_{n_1\ldots n_\nu}^{(\nu)} = \bigotimes_{\alpha=1,\ldots,\nu} A_{n_i} \tag{24}$$

$$= \bigotimes_{\alpha=1,\ldots,\nu} \left(\sum_{k_\alpha \in \mathcal{N}(i)} R_{n_\alpha}(r_{ik_\alpha}, z_{k_\alpha}) \vec{Y}^{ik_\alpha} \right) \tag{25}$$

$$= \sum_{k_1,\ldots,k_\nu} \left[\left(\prod_{\alpha \in 1,\ldots,\nu} R_{n_\alpha}(r_{ik_\alpha}, z_{k_\alpha}) \right) \left(\bigotimes_{\alpha \in 1,\ldots,\nu} \vec{Y}^{ik_\alpha} \right) \right] \tag{26}$$

Comparing Eqs. (23) and (26) it is immediately evident that an Allegro model with N_{layer} layers and an ACE expansion of body order $\nu + 1 = N_{\text{layer}} + 2$ share the core equivariant iterated tensor products $\vec{Y}^{ij} \otimes \vec{Y}^{ik_1} \otimes \ldots \otimes \vec{Y}^{ik_{N_{\text{layer}}}}$. The equivariant Allegro features $\mathbf{V}_n^{ij,L}$ are analogous—but not equivalent—to the full equivariant ACE basis functions $B_{n_1\ldots n_{L+1}}^{(L+1)}$.

The comparison of these expansions of the two models emphasizes, as discussed earlier in the scaling section, that the ACE basis functions carry a full set of n_α indices (which label radial–chemical two-body basis functions), the number of which increases at each iteration, while the Allegro features do not exhibit this increase as a function of the number of layers. This difference is the root of the contrast between the $\mathcal{O}(N_{\text{full-basis}}^{\nu})$ scaling of ACE in the size of the radial–chemical basis $N_{\text{full-basis}}$ and the $\mathcal{O}(1)$ of Allegro. Allegro achieves this more favorable scaling through the learnable channel mixing weights.

A key difference between Allegro and ACE, made clear here, is their differing construction of the scalar pairwise weights. In ACE, the scalar weights carrying ik_α indices are the radial–chemical basis functions R, which are two-body functions of the distance between atoms i and k_α and their chemistry. These correspond in Allegro to the environment embedding weights $w^L_{ik_\alpha,n}$, which—critically—are functions of all the lower-order equivariant features $\mathbf{V}_n^{ij,L'<L}$: the environment embedding weights at layer L are a function of the scalar features from layer $L-1$ (Eq. (14)) which are a function of the equivariant features from layer $L-2$ (Eq. (15)) and so on. As a result, the "pairwise" weights have a hierarchical structure and depend on all previous weights:

$$w^L_{ix,n} = f(\mathbf{V}^{ix,L-1}) \tag{27}$$

$$= f\left(\{w^{L'}_{ix',n'} \text{ for all } n', L' < L, x' \in \mathcal{N}(i)\}\right) \tag{28}$$

where f contains details irrelevant to conveying the existence of the dependence. We hypothesize that this hierarchical nature is in part of why Allegro performs so much better than the ACE model and is a key difference to ACE and its variants, such as NICE. We finally note that the expanded features of Eq. (23)—and thus the final features of any Allegro model—are of finite body order if the environment embedding weights $w^L_{ik_\alpha n}$ are themselves of finite body order. This condition holds if the latent and embedding MLPs are linear. If any of these MLPs contain nonlinearities whose Taylor expansions are infinite, the body order of the environment embedding weights, and thus of the entire model, becomes infinite. Nonlinearities in the two-body MLP are not relevant to the body order and correspond to the use of a nonlinear radial basis in methods such as ACE. Allegro models whose only nonlinearities lie in the two-body embedding MLP are still highly accurate and such a model was used in the experiments on the 3BPA dataset described above.

Discussion

A new type of deep-learning interatomic potential is introduced that combines high prediction accuracy on energies and forces, enabled by its equivariant architecture, with the ability to scale to large system sizes, due to the strict locality of its geometric representations. The Allegro method surpasses the state-of-the-art set by atom-centered message-passing neural network models for interatomic interactions in terms of combined accuracy and scalability. This makes it possible to predict structural and kinetic properties from molecular dynamics simulations of complex systems of millions of atoms at nearly first-principles fidelity.

Our findings enable the study of molecular and materials system with equivariant neural networks that were previously inaccessible and raise broad questions about the optimal choice of representation and learning algorithm for machine learning on molecules and materials. We note that the Allegro method naturally offers a trade-off between accuracy and computational speed, while still offering efficient parallel scalability. Models of higher accuracy can be obtained by choosing networks with higher capacity (including larger numbers of features and more layers), but we also found a small, fast model to work sufficiently well to capture complex structural and kinetic properties in our example applications. It would be of great value to the community to conduct a detailed analysis of this accuracy-speed trade-off across different machine learning interatomic potentials and materials.

The correspondences between the Allegro architecture and the atomic cluster expansion (ACE) formalism also raise questions about how and why Allegro is able to outperform the systematic ACE basis expansion. We speculate that our method's performance is due in part to the learned dependence of the environment embedding weights at each layer on the full scalar latent features from all previous layers. This dependence may allow the importance of an atom to higher body-order interactions to be learned as a function of lower body-order descriptions of its environment. It stands in stark contrast to ACE, where the importance of any higher body-order interaction is learned separately from lower body-order descriptions of the local structure. We believe further efforts to understand this correspondence are a promising direction for future work. Similarly, we believe a systematic study of the completeness of the prescribed architecture will be of high interest.

Another important goal for future work is to obtain a better understanding of when explicit long-range terms are required in machine learning interatomic potentials, how to optimally incorporate them with local models, and to what extent message-passing interatomic potentials may or may not implicitly capture these interactions. For example, it would be interesting to combine the Allegro potential with an explicit long-range energy term. In particular, the strict locality of the Allegro model naturally facilitates separation of the energy into a short-range term and a physically motivated long-range term.

Methods

Software

All experiments were run with the Allegro code available at https://github.com/mir-group/allegro under git commit `a5128c2a86350762215dad6bd8bb42875ebb06cb`. In addition, we used the NequIP code available at https://github.com/mir-group/nequip with version 0.5.3, git commit `eb6f9bca7b36162abf69ebb017049599b4ddb09c`, as well as `e3nn` with version 0.4.4[56], PyTorch with version 1.10.0[57], and Python with version 3.9.7. The LAMMPS experiments were run with the LAMMPS code available at https://github.com/lammps/lammps.git under git commit `9b989b186026c6fe9da354c79cc9b4e152ab03af` with the `pair_allegro` code available at https://github.com/mir-group/pair_allegro, git commit `0161a8a8e2fe0849165de9eeae3fbb987b294079`. The VESTA software was used to generate Fig. 4[58]. Matplotlib was used for plotting results[59].

Reference training sets

revised MD-17. The revised MD-17 dataset consists of ten small organic molecules, for which 100,000 structures were computed at DFT (PBE/def2-SVP) accuracy using a very tight SCF convergence and very dense DFT integration grid[43]. The structures were recomputed from the original MD-17 dataset[10,44,45]. The data can be obtained at https://figshare.com/articles/dataset/Revised_MD17_dataset_rMD17_/12672038. We use 950 structures for training, 50 structures for validation (both sampled randomly), and evaluate the test error on all remaining structures.

3BPA. The 3BPA dataset consists of 500 training structures at $T = 300$ K, and test data at 300 K, 600 K, and 1200 K, of dataset size of 1669, 2138, and 2139 structures, respectively. The data were computed using Density Functioal Theory with the ωB97X exchange-correlation functional and the 6-31G(d) basis set. For details, we refer the reader to[24]. The dataset was downloaded from https://pubs.acs.org/doi/full/10.1021/acs.jctc.1c00647.

QM9. The QM9 data consist of 133,885 structures with up to 9 heavy elements and consisting of species H, C, N, O, F in relaxed geometries.

Structures are provided together with a series of properties computed at the DFT/B3LYP/6-31G(2df,p) level of theory. The dataset was downloaded from https://figshare.com/collections/Quantum_chemistry_structures_and_properties_of_134_kilo_molecules/978904. In line with previous work, we excluded the 3054 structures that failed the geometry consistency check, resulting in 130,831 total structures, of which we use 110,000 for training, 10,000 for validation and evaluate the test error on all remaining structures. Training was performed in units of [eV].

Li_3PO_4. The Li_3PO_4 structure consists of 192 atoms. The reference dataset was obtained from two AIMD simulations both of 50 ps duration, performed in the Vienna Ab-Initio Simulation Package (VASP)[60–62] using a generalized gradient PBE functional[63], projector augmented wave pseudopotentials[64], a plane-wave cutoff of 400 eV and a Γ-point reciprocal-space mesh. The integration was performed with a time step of 2 fs in the NVT ensemble using a Nosé–Hoover thermostat. The first 50 ps of the simulation were performed at $T = 3000$ K in the molten phase, followed by an instant quench to $T = 600$ K and a second 50 ps simulation at $T = 600$ K. The two trajectories were combined and the training set of 10,000 structures as well as the validation set of 1000 were sampled randomly from the combined dataset of 50,000 structures.

Ag. The Ag system is created from a bulk face-centered-cubic structure with a vacancy, consisting of 71 atoms. The data were sampled from AIMD simulations at $T = 1111$ K (90% of the melting temperature of Ag) with Gamma-point k-sampling as computed in VASP using the PBE exchange-correlation functional[60–62]. Frames were then extracted at least 25 fs apart, to limit correlation within the trajectory, and each frame was recalculated with converged DFT parameters. For these calculations, the Brillouin zone was sampled using a $(2 \times 2 \times 3)$ Gamma-centered k-point grid, and the electron density at the Fermi-level was approximated using Methfessel–Paxton smearing[65] with a sigma value of 0.05. A cutoff energy of 520 eV was employed, and each calculation was non-spin-polarized.

Molecular dynamics simulations

Molecular Dynamics simulations were performed in LAMMPS[66] using the pair style `pair_allegro` implemented in the Allegro interface, available at https://github.com/mir-group/pair_allegro. We run the Li_3PO_4 production and timing simulations under an NVT ensemble at $T = 600$ K, using a time step of 2 fs, a Nosé-Hoover thermostat and a temperature damping parameter of 40 time steps. The Ag timing simulations are run also in NVT, at a temperature of $T = 300$ K using a time step of 5 fs, a Nosé-Hoover thermostat and a temperature damping parameter of 40 time steps. The larger systems are created by replicating the original structures of 192 and 71 atoms of Li_3PO_4 and Ag, respectively. We compute the RDF and ADFs for Li_3PO_4 with a maximum distance of 10 Å (RDF) and 2.5 Å (ADFs). We start the simulation from the first frame of the AIMD quench simulation. RDF and ADF for Allegro were averaged over ten runs with different initial velocities, the first 10 ps of the 50 ps simulation were discarded in the RDF/ADF analysis to account for equilibration.

Training details

Models were trained on a NVIDIA V100 GPU in single-GPU training.

revMD-17 and 3BPA. The revised MD-17 models were trained with a total budget of 1000 structures, split into 950 for training and 50 for validation. The 3BPA model was trained with a total budget of 500 structures, split into 450 for training and 50 for validation. The dataset was re-shuffled after each epoch. We use three layers, 128 features for both even and odd irreps and a $\ell_{\max} = 3$. The 2-body latent MLP consists of four hidden layers of dimensions [128, 256, 512, 1024], using SiLU nonlinearities on the outputs of the hidden layers[67]. The later latent MLPs consist of three hidden layers of dimensionality [1024, 1024, 1024] using SiLU nonlinearities for revMD-17 and no nonlinearities for 3BPA. The embedding weight projection was implemented as a single matrix multiplication without a hidden layer or a nonlinearity. The final edge energy MLP has one hidden layer of dimension 128 and again no nonlinearity. All four MLPs were initialized according to a uniform distribution of unit variance. We used a radial cutoff of 7.0 Å for all molecules in the revMD-17 dataset, except for naphthalene, for which a cutoff of 9.0 Å was used, and a cutoff of 5.0 Å for the 3BPA dataset. We have also included an ablation study on the cutoff radius for the large naphthalene molecule which can be found in Supplementary Table 1. We use a basis of eight non-trainable Bessel functions for the basis encoding with the polynomial envelope function using $p = 6$ for revMD-17 and $p = 2$ for 3BPA. We found it particularly important to use a low exponent p in the polynomial envelope function for the 3BPA experiments. We hypothesize that this is due to the fact that a lower exponent provides a stronger decay with increasing interatomic distance (see Supplementary Fig. 1), thereby inducing a stronger inductive bias that atoms j further away from a central atom i should have smaller pair energies E_{ij} and thus contribute less to atom i's site energy E_i. RevMD-17 models were trained using a joint loss function of energies and forces:

$$\mathcal{L} = \frac{\lambda_E}{B}\sum_{b}^{B}\left(\hat{E}_b - E_b\right)^2 + \frac{\lambda_F}{3BN}\sum_{i=1}^{BN}\sum_{\alpha=1}^{3}||-\frac{\partial\hat{E}}{\partial r_{i,\alpha}} - F_{i,\alpha}||^2 \qquad (29)$$

where $B, N, E_b, \hat{E}_b, F_{i,\alpha}$ denote the batch size, number of atoms, batch of true energies, batch of predicted energies, and the force component on atom i in spatial direction α, respectively and λ_E, λ_F are energy and force weights. Following previous works, for the revMD-17 data the force weight was set to 1000 and the weight on the total potential energies was set to 1. For the 3BPA molecules, as in ref. [68], we used a per-atom MSE term that divides the energy term by N_{atoms}^2 because (a) the potential energy is a global size-extensive property, and (b) we use a MSE loss function:

$$\mathcal{L} = \frac{\lambda_E}{B}\sum_{b}^{B}\left(\frac{\hat{E}_b - E_b}{N}\right)^2 + \frac{\lambda_F}{3BN}\sum_{i=1}^{BN}\sum_{\alpha=1}^{3}||-\frac{\partial\hat{E}}{\partial r_{i,\alpha}} - F_{i,\alpha}||^2 \qquad (30)$$

After this normalization, both the energy and the force term receive a weight of 1. Models were trained with the Adam optimizer[69] in PyTorch[57], with default parameters of $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ without weight decay. We used a learning rate of 0.002 and a batch size of 5. The learning rate was reduced using an on-plateau scheduler based on the validation loss with a patience of 100 and a decay factor of 0.8. We use an exponential moving average with weight 0.99 to evaluate on the validation set as well as for the final model. Training was stopped when one of the following conditions was reached: (a) a maximum training time of 7 days, (b) a maximum number of epochs of 100,000, (c) no improvement in the validation loss for 1000 epochs, (d) the learning rate dropped lower than 1e-6. We note that such long wall times are usually not required and highly accurate models can typically be obtained within a matter of hours or even minutes. All models were trained with float32 precision.

3BPA, NequIP. The NequIP models on the 3BPA dataset were trained with a total budget of 500 molecules, split into 450 for training and 50 for validation. The dataset was re-shuffled after each epoch. We use 5 layers, 64 features for both even and odd irreps and a $\ell_{\max} = 3$. We use a radial network of three layers with 64 hidden neurons and SiLU nonlinearities. We further use equivariant, SiLU-based gate nonlinearities as outlined in ref. [15], where even and odd scalars are not gated, but operated on directly by SiLU and tanh nonlinearities, respectively. We

used a radial cutoff of 5.0 Å and a non-trainable Bessel basis of size 8 for the basis encoding with a polynomial envelope function using $p = 2$. Again, a low p value was found to be important. We use again a per-atom MSE loss function in which both the energy and the force term receive a weight of 1. Models were trained with Adam with the AMS-Grad variant in the PyTorch implementation[57,69–71], with default parameters of $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ without weight decay. We used a learning rate of 0.01 and a batch size of 5. The learning rate was reduced using an on-plateau scheduler based on the validation loss with a patience of 50 and a decay factor of 0.8. We use an exponential moving average with weight 0.99 to evaluate on the validation set as well as for the final model. Training was stopped when one of the following conditions was reached: (a) a maximum training time of 7 days, (b) a maximum number of epochs of 100,000, (c) no improvement in the validation loss for 1000 epochs, (d) the learning rate dropped lower than 1e-6. We note that such long wall times are usually not required and highly accurate models can typically be obtained within a matter of hours or even minutes. All models were trained with float32 precision. We use a per-atom shift μ_{Z_i} via the average per-atom potential energy over all training frames and a per-atom scale σ_{Z_i} as the root-mean-square of the components of the forces over the training set.

Li_3PO_4. The Li_3PO_4 model was trained with a total budget of 11,000 structures, split into 10,000 for training and 1000 for validation. The dataset was re-shuffled after each epoch. We use one layer, 1 feature of even parity and $\ell_{\max} = 1$. The 2-body latent MLP consists of 2 hidden layers of dimensions [32, 64], using SiLU nonlinearities[67]. The later latent MLP consist of 1 hidden layer of dimensionality [64], also using a SiLU nonlinearity. The embedding weight projection was implemented as a single matrix multiplication without a hidden layer or a nonlinearity. The final edge energy MLP has one hidden layer of dimension 32 and again no nonlinearity. All four MLPs were initialized according to a uniform distribution of unit variance. We used a radial cutoff of 4.0 Å and a basis of eight non-trainable Bessel functions for the basis encoding with the polynomial envelope function using $p = 48$. The model was trained using a joint loss function of energies and forces. We use again the per-atom MSE as describe above and a weighting of 1 for the force term and 1 for the per-atom MSE term. The model was trained with the Adam optimizer[69] in PyTorch[57], with default parameters of $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ without weight decay. We used a learning rate of 0.001 and a batch size of 1. The learning rate was reduced using an on-plateau scheduler based on the validation loss with a patience of 25 and a decay factor of 0.5. We use an exponential moving average with weight 0.99 to evaluate on the validation set as well as for the final model. Training was stopped when one of the following conditions was reached: (a) a maximum training time of 7 days, (b) a maximum number of epochs of 100,000, (c) no improvement in the validation loss for 1000 epochs, (d) the learning rate dropped lower than 1e-5. The model was trained with float32 precision.

Ag. The Ag model was trained with a total budget of 1000 structures, split into 950 for training and 50 for validation, and evaluated on a separate test set of 159 structures. The dataset was re-shuffled after each epoch. We use 1 layer, 1 feature of even parity and $\ell_{\max} = 1$. The 2-body latent MLP consists of 2 hidden layers of dimensions [16, 32], using SiLU nonlinearities[67]. The later latent MLP consists of 1 hidden layer of dimensionality [32], also using a SiLU nonlinearity. The embedding weight projection was implemented as a single matrix multiplication without a hidden layer or a nonlinearity. The final edge energy MLP has one hidden layer of dimension 32 and again no nonlinearity. All four MLPs were initialized according to a uniform distribution. We used a radial cutoff of 4.0 Å and a basis of eight non-trainable Bessel functions for the basis encoding with the polynomial envelope function using $p = 48$. The model was trained using a joint loss function of energies and forces. We use again the per-atom MSE as describe above and a weighting of 1 for the force term and 1 for the per-atom MSE term. The model was trained with the Adam optimizer[69] in PyTorch[57], with default parameters of $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ without weight decay. We used a learning rate of 0.001 and a batch size of 1. The learning rate was reduced using an on-plateau scheduler based on the validation MAE of the energy with a patience of 25 and a decay factor of 0.8. We use an exponential moving average with weight 0.999 to evaluate on the validation set as well as for the final model. Training was stopped when one of the following conditions was reached: (a) a maximum training time of approximately 14 days, (b) a maximum number of epochs of 100,000, (c) no improvement in the validation loss for 1000 epochs, (d) the learning rate dropped lower than 1e-5. All models were trained with float32 precision. Again, we note that such long wall times are not required to obtain highly accurate models. We subtract the sum of the reference atomic energies and then apply the linear fitting procedure described above using every 100th reference label in the training set.

QM9. We used 110,000 molecular structures for training, 10,000 for validation, and evaluated the test error on all remaining structures, in line with previous approaches[9,27]. We note that Cormorant and EGNN are trained on 100,000 structures, L1Net is trained on 109,000 structures while NoisyNodes is trained on 114,000 structures. To give an estimate of the variability of training as a function of random seed, we report for the U_0 target the mean and sample standard deviation across three different random seeds, resulting in different samples of training set as well as different weight initialization. We report two models, one with three layers and $\ell_{\max} = 2$ and another one with 1 layer and $\ell_{\max} = 3$, both with 256 features for both even and odd irreps. The 1-layer and 3-layer networks have 7,375,237 and 17,926,533 parameters, respectively. The 2-body latent MLP consists of four hidden layers of dimensions [128, 256, 512, 1024], using SiLU nonlinearities[67]. The later latent MLPs consist of three hidden layers of dimensionality [1024, 1024, 1024], also using SiLU nonlinearities. The embedding weight projection was implemented as a single matrix multiplication without a hidden layer or a nonlinearity. The final edge energy MLP has one hidden layer of dimension 128 and again no nonlinearity. All four MLPs were initialized according to a uniform distribution. We used a radial cutoff of 10.0 Å. We use a basis of 8 non-trainable Bessel functions for the basic encoding with the polynomial envelope function using $p = 6$. Models were trained using a MSE loss on the energy with the Adam optimizer[69] in PyTorch[57], with default parameters of $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ without weight decay. In addition, we use gradient clipping by norm with a maximum norm of 100. The dataset was re-shuffled after each epoch. We used a learning rate of 0.001 and a batch size of 16. The learning rate was reduced using an on-plateau scheduler based on the validation MAE of the energy with a patience of 25 and a decay factor of 0.8. We use an exponential moving average with weight 0.999 to evaluate on the validation set as well as for the final model. Training was stopped when one of the following conditions was reached: (a) a maximum training time of approximately 14 days, (b) a maximum number of epochs of 100,000, (c) no improvement in the validation loss for 1000 epochs, (d) the learning rate dropped lower than 1e-5. All models were trained with float32 precision. Again, we note that such long wall times are not required to obtain highly accurate models. We subtract the sum of the reference atomic energies and then apply the linear fitting procedure described above using every 100th reference label in the training set.

Scaling experiments

Scalability across devices is achieved by implementing an Allegro extension to the LAMMPS molecular dynamics code[66]. The local nature of the Allegro model is compatible with the spatial decomposition approach used in LAMMPS and thus all communication between MPI ranks is handled by existing LAMMPS functionality. The Allegro extension simply transforms the LAMMPS neighbor lists into the format required by the Allegro PyTorch model and stores the resulting forces and energies in the LAMMPS data structures. These operations are performed on the GPU and use the Kokkos performance portability library[72] to entirely avoid expensive CPU work or CPU-GPU data transfer. The scaling experiments were performed on NVIDIA DGX A100s on the ThetaGPU cluster at the Argonne Leadership Computing

Facility, where each node contains 8 GPUs and a total of 320 GB of GPU memory. For the Li_3PO_4 simulation, we use a time step of 2 fs, identical to the reference AIMD simulations, float32 precision, and a temperature of $T = 600$ K on the quenched structure, identical to the production simulations used in the quench simulation. For Ag, we use a time step of 5 fs, a temperature of $T = 300$ K and again float32 precision. Simulations were performed for 1000 time steps after initial warm-up.

Atom-density representations

The Atomic Cluster Expansion (ACE) is a systematic scheme for representing local atomic environments in a body-ordered expansion. The coefficients of the expansion of a particular atomic environment serve as an invariant description of that environment. To expand a local atomic environment, the local atomic density is first projected onto a combination of radial basis functions R and spherical harmonic angular basis functions $\vec{Y}$:

$$A_{zn\ell} = \sum_{j \in \mathcal{N}(i)\,\mathbf{s.t.}\,z_j = z} R_{n\ell}(r_{ij})\,\vec{Y}^{m}_{\ell}(\hat{r}_{ij}) \tag{31}$$

where z runs over all atom species in the system, z_j is the species of atom j, $\mathcal{N}(i)$ is the set of all atoms within the cutoff distance of atom i, also known as its "neighborhood", and the n index runs over the radial basis functions. The m index on A is implicit. The basis projection of body order $\nu+1$ is then defined as:

$$B^{(\nu=1)}_{z_1,n_1} = A_{z_1 n_1 \ell_1 = 0} \tag{32}$$

$$B^{(\nu=2)}_{\substack{z_1,z_2,n_1,n_2\\ \ell_1,\ell_2}} = A_{z_1 n_1 \ell_1} \otimes A_{z_2 n_2 \ell_2} \tag{33}$$

$$\ldots \tag{34}$$

$$B^{(\nu)}_{\substack{z_1 \ldots z_\nu, n_1 \ldots n_\nu\\ \ell_1 \ldots \ell_\nu}} = \bigotimes_{\alpha=1,\ldots,\nu} A_{z_\alpha n_\alpha \ell_\alpha}. \tag{35}$$

Only tensor products outputting scalars—which are invariant, like the final target total energy—are retained here. For example, in Eq. (33), only tensor products combining basis functions inhabiting the same rotation order $\ell_1 = \ell_2$ can produce scalar outputs. The final energy is then fit as a linear model over all the scalars B up to some chosen maximum body order $\nu+1$.

It is apparent from Eq. (35) that a core bottleneck in the Atomic Cluster Expansion is the polynomial scaling of the computational cost of evaluating the B terms with respect to the total number of two-body radial–chemical basis functions $N_{\text{full-basis}}$ as the body order $\nu+1$ increases: $\mathcal{O}(N^{\nu}_{\text{full-basis}})$. In the basic ACE descriptor given above, $N_{\text{full-basis}} = N_{\text{basis}} \times S$ is the number of radial basis functions times the number of species. Species embeddings have been proposed for ACE to remove the direct dependence on S[73]. It retains, however, the $\mathcal{O}(N^{\nu}_{\text{full-basis}})$ scaling in the dimension of the embedded basis $N_{\text{full-basis}}$. NequIP and some other existing equivariant neural networks avert this unfavorable scaling by only computing tensor products of a more limited set of combinations of input tensors. The NICE framework[74] is an idea closely related to ACE that aims to solve the problem of increasing numbers of features by selecting only certain features at each iteration based on principal component analysis.

Normalization

Internal normalization. The normalization of neural networks' internal features is known to be of great importance to training. In this work we follow the normalization scheme of the `e3nn` framework[75], in which the initial weight distributions and normalization constants are chosen so that all components of the network produce outputs that element-wise have approximately zero mean and unit variance. In particular, all sums over multiple features are normalized by dividing by the square root of the number of terms in the sum, which follows from the simplifying assumption that the terms are uncorrelated and thus that their variances add. Two consequences of this scheme that merit explicit mention are the normalization of the embedded environment and atomic energy. Both the embedded environment (Eq. (4)) and atomic energy (Eq. (6)) are sums over all neighbors of a central atom. Thus we divide both by $\sqrt{\langle |\mathcal{N}(i)| \rangle}$ where $\langle |\mathcal{N}(i)| \rangle$ is the average number of neighbors over all environments in the entire training dataset.

Normalization of targets. We found the normalization of the targets, or equivalently the choice of final scale and shift parameters for the network's predictions (see Eq. (5)), to be of high importance. For systems of fixed chemical composition, our default initialization is the following: μ_Z is set for all species Z to the average per-atom potential energy over all training frames $\left\langle \frac{E_{\text{config}}}{N} \right\rangle$; σ_Z is set for all species Z to the root-mean-square of the components of the forces on all atoms in the training dataset. This scheme ensures size extensivity of the potential energy, which is required if one wants to evaluate the potential on systems of different size than what it was trained on. We note that the widely used normalization scheme of subtracting the mean total potential energy across the training set violates size extensivity.

For systems with varying chemical composition, we found it helpful to normalize the targets using a linear pre-fitting scheme that explicitly takes into account the varying chemical compositions: μ_Z is computed by $[N_{\text{config},Z}]^{-1}[E_{\text{config}}]$, where $[N_{\text{config},Z}]$ is a matrix containing the number of atoms of each species in the reference structures, and $[E_{\text{config}}]$ is a vector of reference energies. Details of the normalization calculations and the comparison between different schemes can be found in ref. [68].

Reporting summary

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

Data availability

The Li_3PO_4 and Ag data generated in this study have been deposited in the MaterialsCloud database at https://archive.materialscloud.org/record/2022.128. The revMD-17, 3BPA, and QM9 datasets are publicly available (see "Methods").

Code availability

An open-source software implementation of Allegro is available at https://github.com/mir-group/allegro together with an implementation of the LAMMPS software interface at https://github.com/mir-group/pair_allegro.

References

1. Richards, W. D. et al. Design and synthesis of the superionic conductor na 10 snp 2 s 12. *Nat. Commun.* **7**, 1–8 (2016).
2. Lindorff-Larsen, K., Piana, S., Dror, R. O. & Shaw, D. E. How fast-folding proteins fold. *Science* **334**, 517–520 (2011).
3. Blank, T. B., Brown, S. D., Calhoun, A. W. & Doren, D. J. Neural network models of potential energy surfaces. *J. Chem. Phys.* **103**, 4129–4137 (1995).
4. Handley, C. M., Hawe, G. I., Kell, D. B. & Popelier, P. L. Optimal construction of a fast and accurate polarisable water potential based on multipole moments trained by machine learning. *Phys. Chem. Chem. Phys.* **11**, 6365–6376 (2009).
5. Behler, J. & Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **98**, 146401 (2007).

6. Bartók, A. P., Payne, M. C., Kondor, R. & Csányi, G. Gaussian approximation potentials: the accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.* **104**, 136403 (2010).
7. 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 (2015).
8. Shapeev, A. V. Moment tensor potentials: a class of systematically improvable interatomic potentials. *Multiscale Model. Simul.* **14**, 1153–1173 (2016).
9. Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A. & Müller, K.-R. Schnet–a deep learning architecture for molecules and materials. *J. Chem. Phys.* **148**, 241722 (2018).
10. Chmiela, S., Sauceda, H. E., Müller, K.-R. & Tkatchenko, A. Towards exact molecular dynamics simulations with machine-learned force fields. *Nat. Commun.* **9**, 3887 (2018).
11. Unke, O. T. & Meuwly, M. Physnet: a neural network for predicting energies, forces, dipole moments, and partial charges. *J. Chem. Theory Comput.* **15**, 3678–3693 (2019).
12. Drautz, R. Atomic cluster expansion for accurate and transferable interatomic potentials. *Phys. Rev. B* **99**, 014104 (2019).
13. Christensen, A. S., Bratholm, L. A., Faber, F. A. & Anatole von Lilienfeld, O. Fchl revisited: faster and more accurate quantum machine learning. *J. Chem. Phys.* **152**, 044107 (2020).
14. Gasteiger, J., Groß, J. & Günnemann, S. Directional message passing for molecular graphs. *International Conference on Learning Representations*, Preprint at https://arxiv.org/abs/2003.03123 (2020).
15. Batzner, S. et al. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nat. Commun.* **13**, 1–11 (2022).
16. Mailoa, J. P. et al. A fast neural network approach for direct covariant forces prediction in complex multi-element extended systems. *Nat. Mach. Intell.* **1**, 471–479 (2019).
17. Park, C. W. et al. Accurate and scalable graph neural network force field and molecular dynamics with direct force architecture. *npj Comput. Mater.* **7**, 73 (2021).
18. Xie, Y., Vandermause, J., Sun, L., Cepellotti, A. & Kozinsky, B. Bayesian force fields from active learning for simulation of inter-dimensional transformation of stanene. *npj Comput. Mater.* **7**, 1–10 (2021).
19. Xie, Y. et al. Uncertainty-aware molecular dynamics from bayesian active learning: phase transformations and thermal transport in SiC. Preprint at https://arxiv.org/abs/2203.03824 (2022).
20. Zhang, L., Han, J., Wang, H., Car, R. & Weinan, E. Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. *Phys. Rev. Lett.* **120**, 143001 (2018).
21. Vandermause, J. et al. On-the-fly active learning of interpretable Bayesian force fields for atomistic rare events. *npj Comput. Mater.* **6**, 1–11 (2020).
22. Vandermause, J., Xie, Y., Lim, J. S., Owen, C. & Kozinsky, B. Active learning of reactive Bayesian force fields: application to heterogeneous catalysis dynamics of H/Pt. *Nat. Commun.* **15**, 5183 (2021).
23. Anderson, B., Hy, T. S. & Kondor, R. Cormorant: covariant molecular neural networks. *Adv. Neural Inf. Process. Syst.* **32**, 14537–14546 (2019).
24. Kovács, D. P. et al. Linear atomic cluster expansion force fields for organic molecules: beyond rmse. *J. Chem. Theory Comput.* **17**, 7696–7711 (2021).
25. Schütt, K. et al. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Adv. Neural Inf. Process. Syst.* **30**, 991–1001 (2017).
26. Qiao, Z. et al. Unite: Unitary n-body tensor equivariant network with applications to quantum chemistry. Preprint at https://arxiv.org/pdf/2105.14655 (2021).
27. Schütt, K. T., Unke, O. T. & Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. *International Conference on Machine Learning*, 9377–9388 (PMLR, 2021).
28. Jia, W. et al. Pushing the limit of molecular dynamics with ab initio accuracy to 100 million atoms with machine learning. in *SC20: International Conference For High Performance Computing, Networking, Storage and Analysis*, 1–14 (IEEE, 2020).
29. Lu, D. et al. 86 pflops deep potential molecular dynamics simulation of 100 million atoms with ab initio accuracy. *Comput. Phys. Commun.* **259**, 107624 (2021).
30. Guo, Z. et al. Extending the limit of molecular dynamics with ab initio accuracy to 10 billion atoms. *Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, 205–218 (2022).
31. Nguyen-Cong, K. et al. Billion atom molecular dynamics simulations of carbon at extreme conditions and experimental time and length scales. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 1–12 (Association for Computing Machinery, 2021).
32. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural message passing for quantum chemistry. *International conference on machine learning*, 1263–1272 (PMLR, 2017).
33. Thomas, N. et al. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. Preprint at https://arxiv.org/pdf/1802.08219.pdf (2018).
34. Weiler, M., Geiger, M., Welling, M., Boomsma, W. & Cohen, T. S. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. *Adv. Neural Inf. Process. Syst.* **31**, 10381–10392 (2018).
35. Kondor, R. N-body networks: a covariant hierarchical neural network architecture for learning atomic potentials. Preprint at https://arxiv.org/abs/1803.01588 (2018).
36. Kondor, R., Lin, Z. & Trivedi, S. Clebsch–gordan nets: a fully Fourier space spherical convolutional neural network. *Adv. Neural Inf. Process. Syst.* **31**, 10117–10126 (2018).
37. Haghighatlari, M. et al. Newtonnet: a Newtonian message passing network for deep learning of interatomic potentials and forces. *Digital Discovery* **1**, 333–343 (2022).
38. Thölke, P. & De Fabritiis, G. Torchmd-net: equivariant transformers for neural network based molecular potentials. Preprint at https://arxiv.org/abs/2202.02541 (2022).
39. Brandstetter, J., Hesselink, R., van der Pol, E., Bekkers, E. J. & Welling, M. Geometric and physical quantities improve E(3) equivariant message passing. *International Conference on Learning Representations*, Preprint at https://arxiv.org/abs/2110.02905 (2021).
40. Satorras, V. G., Hoogeboom, E. & Welling, M. E (n) equivariant graph neural networks. In *International Conference on Machine Learning*, 9323–9332 (PMLR, 2021).
41. Bartók, A. P., Kondor, R. & Csányi, G. On representing chemical environments. *Phys. Rev. B* **87**, 184115 (2013).
42. 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*, 770–778 (IEEE, 2016).
43. Christensen, A. S. & von Lilienfeld, O. A. On the role of gradients for machine learning of molecular energies and forces. *Mach. Learn.: Sci. Technol.* **1**, 045018 (2020).
44. Chmiela, S. et al. Machine learning of accurate energy-conserving molecular force fields. *Sci. Adv.* **3**, e1603015 (2017).
45. Schütt, K. T., Arbabzadah, F., Chmiela, S., Müller, K. R. & Tkatchenko, A. Quantum-chemical insights from deep tensor neural networks. *Nat. Commun.* **8**, 13890 (2017).
46. Wang, J., Wolf, R. M., Caldwell, J. W., Kollman, P. A. & Case, D. A. Development and testing of a general amber force field. *J. Comput. Chem.* **25**, 1157–1174 (2004).
47. Wang, L.-P., Chen, J. & Van Voorhis, T. Systematic parametrization of polarizable force fields from quantum chemistry data. *J. Chem. Theory Comput.* **9**, 452–460 (2013).

48. Smith, J. S., Isayev, O. & Roitberg, A. E. Ani-1: an extensible neural network potential with dft accuracy at force field computational cost. *Chem. Sci.* **8**, 3192–3203 (2017).
49. Devereux, C. et al. Extending the applicability of the ani deep learning molecular potential to sulfur and halogens. *J. Chem. Theory Comput.* **16**, 4192–4202 (2020).
50. Ramakrishnan, R., Dral, P. O., Rupp, M. & Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **1**, 1–7 (2014).
51. Yu, X., Bates, J. B., Jellison, G. E. & Hart, F. X. A stable thin-film lithium electrolyte: lithium phosphorus oxynitride. *J. Electrochem. Soc.* **144**, 524–532 (1997).
52. Westover, A. S. et al. Plasma synthesis of spherical crystalline and amorphous electrolyte nanopowders for solid-state batteries. *ACS Appl. Mater. Interfaces* **12**, 11570–11578 (2020).
53. Kalnaus, S., Westover, A. S., Kornbluth, M., Herbert, E. & Dudney, N. J. Resistance to fracture in the glassy solid electrolyte lipon. *J. Mater. Res.* **36**, 787–796 (2021).
54. Li, W., Ando, Y., Minamitani, E. & Watanabe, S. Study of li atom diffusion in amorphous li3po4 with neural network potential. *J. Chem. Phys.* **147**, 214106 (2017).
55. Fuchs, F., Worrall, D., Fischer, V. & Welling, M. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Adv. Neural Inf. Process. Syst.* **33**, 1970–1981 (2020).
56. Geiger, M. & Smidt, T. e3nn: Euclidean neural networks. Preprint at https://arxiv.org/abs/2207.09453 (2022).
57. Paszke, A. et al. Pytorch: an imperative style, high-performance deep learning library. in *Adv. Neural Inf. Process. Syst.* **32**, 8026–8037 (2019).
58. Momma, K. & Izumi, F. Vesta: a three-dimensional visualization system for electronic and structural analysis. *J. Appl. Crystallogr.* **41**, 653–658 (2008).
59. Hunter, J. D. Matplotlib: a 2d graphics environment. *Comput. Sci. Eng.* **9**, 90–95 (2007).
60. Kresse, G. & Hafner, J. Ab initiomolecular dynamics for liquid metals. *Phys. Rev. B* **47**, 558–561 (1993).
61. Kresse, G. & Furthmüller, J. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Comput. Mater. Sci.* **6**, 15–50 (1996).
62. Kresse, G. & Furthmüller, J. Efficient iterative schemes forab initiototal-energy calculations using a plane-wave basis set. *Phys. Rev. B* **54**, 11169–11186 (1996).
63. Perdew, J. P., Burke, K. & Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.* **77**, 3865–3868 (1996).
64. Kresse, G. & Joubert, D. From ultrasoft pseudopotentials to the projector augmented-wave method. *Phys. Rev. B* **59**, 1758–1775 (1999).
65. Methfessel, M. & Paxton, A. High-precision sampling for Brillouin-zone integration in metals. *Phys. Rev. B* **40**, 3616 (1989).
66. Thompson, A. P. et al. LAMMPS—a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comp. Phys. Comm.* **271**, 108171 (2022).
67. Hendrycks, D. & Gimpel, K. Gaussian error linear units (gelus). Preprint at https://arxiv.org/abs/1606.08415 (2016).
68. Sun, L., Batzner, S., Musaelian, A., Yu, X. & Kozinsky, B. On the normalization of potential energies for neural-network-based interatomic potentials training. (2023).
69. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. Preprint at https://arxiv.org/abs/1412.6980 (2014).
70. Loshchilov, I. & Hutter, F. Decoupled weight decay regularization. *International Conference on Learning Representations*. Preprint at https://arxiv.org/abs/1711.05101 (2017).
71. Reddi, S. J., Kale, S. & Kumar, S. On the convergence of adam and beyond. *International Conference on Learning Representations*. Preprint at https://arxiv.org/abs/1904.09237 (2019).
72. Carter Edwards, H., Trott, C. R. & Sunderland, D. Kokkos: enabling manycore performance portability through polymorphic memory access patterns. *J. Parallel Distrib. Comput.* **74**, https://www.osti.gov/biblio/1106586 (2014).
73. Darby, J. P., Kermode, J. R. & Csányi, G. Compressing local atomic neighbourhood descriptors. *npj Comput. Mater.* **8**, 166 (2022).
74. Nigam, J., Pozdnyakov, S. & Ceriotti, M. Recursive evaluation and iterative contraction of n-body equivariant features. *J. Chem. Phys.* **153**, 121101 (2020).
75. Geiger, M., & Smidt, T. e3nn: Euclidean neural networks. arXiv preprint https://doi.org/10.48550/arXiv.2207.09453 (2022).
76. Gasteiger, J., Becker, F. & Günnemann, S. Gemnet: Universal directional graph neural networks for molecules. *Adv. Neural Inf. Process. Syst.* **34**, 6790–6802 (2021).
77. Gasteiger, J., Giri, S., Margraf, J. T. & Günnemann, S. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. Preprint at https://arxiv.org/abs/2011.14115 (2020).
78. Finzi, M., Stanton, S., Izmailov, P. & Wilson, A. G. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. in *International Conference on Machine Learning*, 3165–3176 (PMLR, 2020).
79. Miller, B. K., Geiger, M., Smidt, T. E. & No'e, F. Relevance of rotationally equivariant convolutions for predicting molecular properties. Preprint at https://arxiv.org/abs/2008.08461 (2020).
80. Liu, Y. et al. Spherical message passing for 3d graph networks. Preprint at https://arxiv.org/abs/2102.05013 (2021).
81. Godwin, J. et al. Simple gnn regularisation for 3d molecular property prediction and beyond. In *International Conference on Learning Representations* (2021).

Acknowledgements

Authors thank Dr. Nicola Molinari for the helpful discussions. This work was supported primarily by the US Department of Energy. S.B., A.J., and B.K. were supported by DOE Office of Basic Energy Sciences Award No. DE-SC0022199. L.S. and B.K. were supported by the Integrated Mesoscale Architectures for Sustainable Catalysis (IMASC), an Energy Frontier Research Center, Award No. DE-SC0012573. B.K. acknowledges partial support from NSF through the Harvard University Materials Research Science and Engineering Center Grant No. DMR-2011754 and Bosch Research. A.M. is supported by U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Computational Science Graduate Fellowship under Award Number(s) DE-SC0021110. C.J.O. is supported by the National Science Foundation Graduate Research Fellowship Program, Grant No. DGE1745303. Work at Bosch Research by M.K. was partially supported by ARPA-E Award No. DE-AR0000775 and used resources of the Oak Ridge Leadership Computing Facility at Oak Ridge National Laboratory, which is supported by the Office of Science of the Department of Energy under Contract DE-AC05-00OR22725. The authors acknowledge computing resources provided by the Harvard University FAS Division of Science Research Computing Group and by the Texas Advanced Computing Center (TACC) at The University of Texas at Austin under allocation DMR20013. This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

Author contributions

S.B. and A.M. jointly conceived the Allegro model architecture, derived the theoretical analysis of the model, and wrote the first version of the manuscript. A.M. implemented the software and contributed to running experiments. S.B. originally proposed to work on an architecture that can capture many-body information without atom-centered message passing, conducted the experiments, and contributed to the software implementation. A.J. wrote the LAMMPS interface, including parallelization across devices. L.S. proposed the linear fitting for the per-species

initialization and implemented it. C.J.O. generated the Ag data. M.K. generated the AIMD dataset of Li_3PO_4, wrote software for the analysis of MD results, and contributed to the analysis of results on this system. B.K. supervised and guided the project from conception to design of experiments, implementation, theory, as well as analysis of data. All authors contributed to the manuscript.

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/s41467-023-36329-y.

Correspondence and requests for materials should be addressed to Simon Batzner or Boris Kozinsky.

Peer review information *Nature Communications* thanks Huziel Sauceda, Claudio Zeni and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permissions information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/.

© The Author(s) 2023