

<https://doi.org/10.1038/s41524-025-01541-5>

MGNN: Moment Graph Neural Network for Universal Molecular Potentials



Jian Chang & Shuze Zhu

The quest for efficient and robust deep learning models for molecular systems representation is increasingly critical in scientific exploration. The advent of message passing neural networks has marked a transformative era in graph-based learning, particularly in the realm of predicting chemical properties and expediting molecular dynamics studies. We present the Moment Graph Neural Network (MGNN), a rotation-invariant message passing neural network architecture that capitalizes on the moment representation learning of 3D molecular graphs, is adept at capturing the nuanced spatial relationships inherent in three-dimensional molecular structures. From benchmark tests on public datasets, MGNN delivers multiple state-of-the-art results on QM9, revised MD17 and MD17-ethanol. Its generalizability and efficiency are also tested in additional systems including 3BPA and 25-element high-entropy alloys. The prowess of MGNN also extends to dynamic simulations, accurately predicting the structural and kinetic properties of complex systems such as amorphous electrolytes, with results that closely align with those from ab-initio simulations. The application of MGNN to the simulation of molecular spectra exemplifies its potential to offer a promising alternative to traditional electronic structure methods.

The computational determination of molecular properties has been significantly accelerated by machine learning. The crux of this advancement lies in the development of machine learning potentials with remarkable efficiency and fidelity. Although the usage of local descriptors in most machine learning interatomic potentials have improved parallel computation and scalability^{1–6}, yet they fall short in accuracy compared with atom-centered message-passing neural network (MPNNs) interaction potentials within the concept of graph neural network (GNN)^{7–12}, where the molecular systems are treated as 3D graphs, with atoms as nodes linked by edges within a defined cutoff radius. These networks have achieved groundbreaking results, often surpassing traditional models with minimal manual feature engineering. They encode atoms into a high-dimensional space and model interatomic interactions through message passing, effectively serving as surrogates for quantum mechanical simulations^{7–16}. Machine-learning interatomic potentials have provided promising solutions to bridge the gap between expensive electronic structure methods and efficient classical interatomic potentials^{17–20}. Most recently, GNN-based machine-learning interatomic potentials trained on the periodic table (for example, M3GNet²¹ and CHGNet¹⁷) have demonstrated the possibility of universal interatomic potentials that may not require chemistry-specific training for each new application^{17,21–23}.

In principle, to accurately predict scalar quantities like potential energy, vectorial properties such as forces and dipole moments, and even tensorial

properties like polarizabilities^{24–27}, a GNN which fully accommodates the transformation properties of scalar or tensor is required. For instance, while the translation and rotation of a small molecule do not alter its formation energy, the forces acting on its atoms rotate with the molecule. Most available GNN machine learning potentials address above points by empirically constructing neural network using geometrical quantities like interatomic-distances, bond angles^{28–31}.

To learn molecular properties satisfying all the needed symmetries, systematical and rigorous routines to treat symmetries are highly desirable. Currently, many machine learning potentials use equivariant message-passing frameworks to achieve such goals, which typically use spherical harmonics as angular basis functions, followed by Clebsch-Gordan contraction to maintain rotational symmetry^{11,12,32–37}. However, these calculations are usually expensively. To address such issue, there are emerging efforts on finding efficient computational methods. For instance, a mathematically equivalent and simple message-passing framework that circumvents spherical harmonics is recently proposed, which avoids the usage of high-dimensional Clebsch-Gordan coupling matrix^{26,38,39}. On the other hand, there are a series of mathematical theorems which establishes that the Moment Tensor Potentials⁴ can provably approximate any regular function satisfying all the needed symmetries, while simultaneously being computationally efficient.

Aiming to develop simple and efficient GNN to learn molecular properties satisfying symmetry requirement, as inspired by Moment Tensor Potential⁴, we introduce the Moment Graph Neural Network (MGNN), which innovatively propagates information between molecular systems using moments—quantities that encapsulate the spatial relationships between atoms. Our framework defines molecular moments⁴ to convey relative spatial relationships between atoms within a molecule, utilizing Chebyshev polynomials to encode interatomic distance information instead of more commonly used functions like Bessel and Gaussian radial basis functions^{28,40}. Impressively, MGNN demonstrates multiple state-of-the-art results on benchmark datasets including QM9^{41,42}, revised MD17⁴³, and MD17-ethanol⁴⁴. Its generalizability and efficiency are also tested in additional systems including 3BPA⁴⁵ and 25-element high-entropy alloys⁴⁶. Additionally, we demonstrate the high consistency of our framework in predicting molecular dynamics simulations of amorphous electrolytes. Moreover, MGNN's predictive capabilities extend to the calculation of dipole moments and polarizabilities for organic molecules, such as ethanol, enabling the rapid simulation of molecular spectra in vacuum conditions.

Results

The MGNN architecture

Overview. The Hamiltonian of a molecular system is uniquely determined by the external potential, depending on the charge quantities $\{Z_i\}$ and atomic positions $\{\vec{r}_i\}$. Formally, a molecule in a specific conformation can be represented as a point cloud with n atoms denoted as $M = (Z, R)$, where $Z = (Z_1, \dots, Z_n) \in \mathbb{Z}^n$ represents the atom type vector and $R = (r_1, \dots, r_n) \in \mathbb{R}^{n \times 3}$ represents the atom coordinate matrix.

MGNNs treat molecules as graphs, with atoms as nodes, and edges representing connections between atoms, either based on predefined molecular graphs or atoms connected within a cutoff distance r_{cut} . MGNNs represent each atom through atomic embeddings $h_i \in \mathbb{R}^{H_a}$ where H_a is the number of atom features and the edges between nodes i and j through edge embeddings $e_{ij} \in \mathbb{R}^{H_e}$ where H_e is the number of edge features. These embeddings are updated in each layer through information propagation. In general, the Message Passing Neural Networks (MPNN)¹⁴ framework achieves message passing by $m_v^{t+1} = \sum_{w \in \mathcal{N}(v)} M_t(h_v^t, h_w^t, e_{vw})$ and updates by $h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$, where M_t represents the message function, U_t represents the vertex update function, and $\mathcal{N}(v) = \{w | \|\vec{r}_{vw}\| \leq r_{cut}\}$ denotes the neighbors of node v in graph G . Our MGNN framework utilizes moment interaction to construct M_t and U_t .

In this work, a molecular graph is represented as $G = (A, E, T)$, where A denotes the atoms (nodes), E the edges within cutoff r_{cut} , and T the triplets. Each atom i forms an edge with every neighboring atom within its cutoff distance, as demonstrated in Fig. 1a, constructing edges ij , ik , and corresponding triplets ijk , ijm , ikm to formulate the moments. Within a triplet composed of nodes i, j, k and edges ij , ik , the message of triplets are used to update edge representation features, which are further used to update node features. As shown in Fig. 1b, for a triplet ijk whose center is atom i and edges are ij and ik , the information within this triplet first passes to edge ij (symmetrically to ik), and then the message of edge ij passes to the central node i .

The MGNN framework comprises an embedding block, several sequential interaction layers, and an output block, as shown in Fig. 1c. The molecular charge quantities (labeled as Z) first pass through the embedding block to obtain the initial node representation vectors (labeled as X). These initial vectors, along with the positions of atoms in Cartesian space containing geometry information (labeled as R), are then input into an interaction layer, which uses an interaction block that generates a correction (ΔX) to be added to the input (X) to generate the current layer output (i.e., $X + \Delta X$), which serves as the input for the next interaction layer or output layer. The interaction block (detailed in Fig. 1d) captures how atoms interact with each other, by processing the

message from the neighbors of atom i , during which the interaction between edges from atom i (i.e., triplet moment interaction as detailed in Fig. 1e) is involved. There are N sequential interaction layers in total. The output after the last interaction layer is labeled as X^L , which is then fed into target-dependent output block (detailed in Fig. 2). By aggregating the message of triplets, we construct a comprehensive representation of the molecular structure within the graph, which is then propagated through the interaction layers to update atomic representations. The resulting feature vectors for each atom are fed into the output block, tailored to predict specific properties, culminating in the model's final predictions. The output module of the MGNN model is adept at generating scalar, vector, and tensor outputs, demonstrating its versatility in capturing and predicting a wide range of molecular properties.

The definition of MLP in current work (as shown in Figs. 1 and 2) is $MLP(x) = W^{out}(\sigma(W^{in}x + b^{in})) + b^{out}$, where W^{in} and W^{out} are learnable weights and b^{in} and b^{out} are learnable biases, and $\sigma(x)$ denotes the element-wise nonlinear activation function. In our experiments, we utilize the SiLU^{47,48} and Mish⁴⁹ activation functions. The action of “Concat”, “Reshape”, “Split” in current work (as shown in Figs. 1 and 2) are used to manipulate data dimensions. We use *torch.split* for “Split” operation. Given a tensor A , suppose the last dimension of A has size $2F$. By performing a split along the last dimension, we can divide A into two chunks, each with a last dimension of size F , while keeping the sizes of other dimensions unchanged. Of course, it's also possible to split the tensor into other proportions, such as 0.5 F and 1.5 F , for more details, refer to the PyTorch⁵⁰ documentation. All the notations “ W ” and “ b ” with or without superscript are learnable parameters and we will hint their shape when necessary. Next, we will provide a detailed introduction to each block. The notions employed in this paper are summarized in Supporting Information S1.

Embedding. MGNN works on embedded feature, which is initialized by the embedding block shown in Fig. 1c. Throughout interaction layers, we represent the atoms using a tuple of features $X^l = (x_1^l, \dots, x_n^l)$, with $x_i^l \in \mathbb{R}^F$ with the number of feature maps F , the number of atoms n and the current interaction layer l . We maintain a constant number of feature maps at $F = 512$ throughout the network, if not stated otherwise. The representation of atom i is initialized using an embedding depend on the atom type $Z_i \in \mathbb{N}$:

$$x_i^0 = \text{Embedding}(Z_i) \quad (1)$$

The atom type embeddings are learned during training.

Interaction layer. Here we explain the interaction layer shown in Fig. 1c by implementing Eqs. (2)–(13). Within the interaction layer, an interaction block (as depicted in Fig. 1d) produces a correction $\Delta X = \{\Delta x_i\}$ that is used to update the input embedded feature $X = \{x_i\}$. Formally, the update uses a residual connection inspired by ResNet⁵¹, where l is the layer index, and $\mathcal{N}(i)$ is the neighbor list of atom i that determines which edges and angles are involved in message passing.

$$x_i^{l+1} = x_i^l + \Delta x_i^l(x_i^l, \mathcal{N}(i)) \quad (2)$$

The residual Δx_i^l is computed through integrating information from edges and angles associated with $\mathcal{N}(i)$. For example, for the $\mathcal{N}(i)$ shown in Fig. 1a, one angular message is obtained from two edge vectors \vec{r}_{ij} and \vec{r}_{ik} originating from node i to node j and k within one triplet ijk , another two angular messages are from triplets ijm and imk . Since each interaction layer has same structure, for clarity, we no longer indicate the layer index l unless necessary.

In one interaction block (as depicted in Fig. 1d), the atomistic representations obtained from the preceding layer (embedding block or previous interaction layer) are passed through a learnable MLP layer:

$$h_i = MLP(x_i) \quad (3)$$

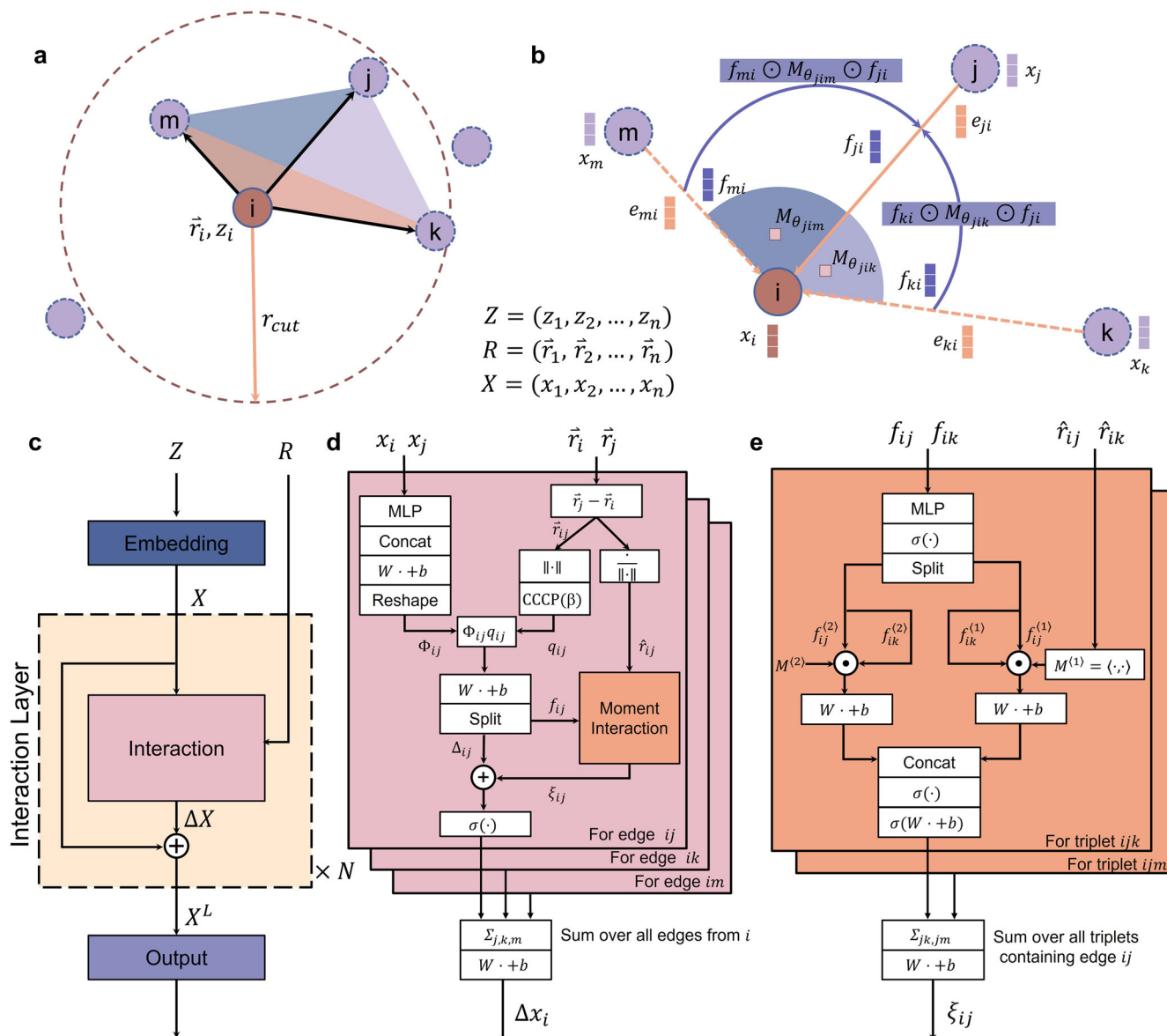


Fig. 1 | The moment message passing scheme and the architecture of MGNN. **a** Atom i receives information from neighboring atoms j, k , and m within a specified cutoff radius r_{cut} , forming three distinct triplets (ijk, ijm, ikm) with i as the central atom. **b** Information flow within a specific triplet ijk , where i serves as the central atom and ij and jk constitute the connecting edges. **c** The sequential arrangement of MGNN, from the initial embedding module, through multiple (N) interaction layers to the output module. **d** Architecture of interaction layer, which contains moment

interaction layer. **e** Architecture of moment interaction layer. “Concat” refers to the concatenation of two feature vectors, “CCCC” the continuous cutoff and Chebyshev polynomials-based radial basis function, “Split” the splitting of a long vector into shorter vectors, “ \oplus ” the residual connection, “ \odot ” the element-wise product, and the dot “ \cdot ” in the brackets or follows the “ W ” denotes the input from previous block. See main text for detailed description.

After updating the atomistic representations using Eq. (3), we concatenate the feature maps of the source and destination nodes of each edge (labeled as $h_i || h_j$ where the symbol “ $||$ ” denotes the concatenation of two vectors) to obtain vectors of dimension $2F$. Subsequently, we pass them through a linear layer to convert the feature size of each edge to dimension βF , and then reshape them into a matrix $\Phi_{ij} \in \mathbb{R}^{F \times \beta}$ (Eq. (4)) and next perform matrix multiplication (Eq. (6)) with a vector $q_{ij} \in \mathbb{R}^\beta$ (Eq. (5)) which encodes the information of r_{ij} using Chebyshev polynomials. The outcome is an edge feature vector $e_{ij} \in \mathbb{R}^F$ (Eq. (6)):

$$\Phi_{ij} = \text{Reshape}(W(h_i || h_j) + b) \in \mathbb{R}^{F \times \beta} \quad (4)$$

$$q_{ij} = \psi^{(\beta)}(r_{ij}) \in \mathbb{R}^\beta \quad (5)$$

$$e_{ij} = \Phi_{ij} q_{ij} \in \mathbb{R}^F \quad (6)$$

where $W \in \mathbb{R}^{\beta F \times 2F}$ $b \in \mathbb{R}^{\beta F}$.

The $\psi^{(\beta)}$ in Eq. (5) is the continuous cutoff and Chebyshev polynomials-based radial basis function (“CCCC(β)” in Fig. 1d), denoted as

$$\psi^{(\beta)}(r_{ij}) = \begin{cases} C^{(\beta)}(r_{ij})(r_{ij} - r_{cut})^2, & r_{ij} < r_{cut} \\ 0, & r_{ij} \geq r_{cut} \end{cases} \quad (7)$$

Here $C^{(\beta)}$ are Chebyshev polynomials on the interval $[r_{min}, r_{cut}]$, where r_{min} is the minimal distance between atoms and is usually set to zero, r_{cut} is the cutoff radius which is introduced to ensure a smooth behavior of MGNN when atoms leave or enter the interaction neighborhood.

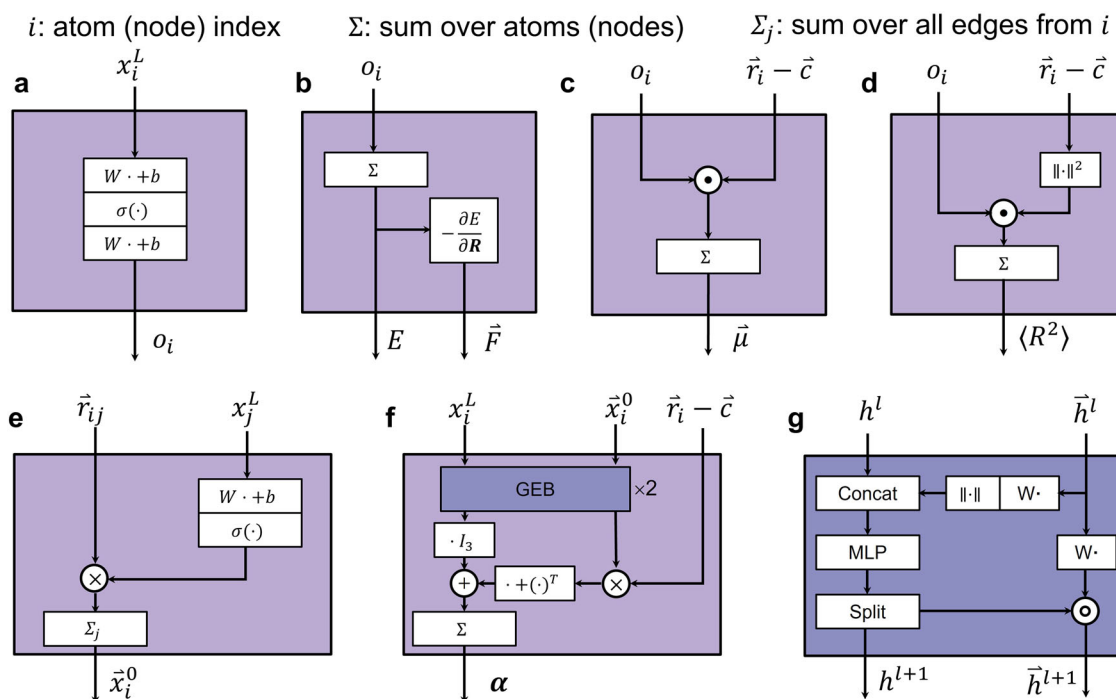


Fig. 2 | Output blocks. **a** The MLP block. **b** The energy and force. **c** Dipole moment. **d** Electronic spatial extent. **e** Vectorial features initialization. **f** Polarizability, the “GEB×2” denote there are two gated equivariant blocks (GEBs). **g** The gated equivariant block.

An illustration of the radial basis functions is given in Fig. S1 (Supporting Information S2).

Once the feature maps e_{ij} on the edge are obtained, we split its information (Eq. (8)) to prepare the update on the features of atoms and edges.

$$\Delta_{ij}, f_{ij} = \text{Split}\left(We_{ij} + b\right) \quad (8)$$

where $W \in \mathbb{R}^{3F \times F}$, $b \in \mathbb{R}^{3F}$, $\Delta_{ij} \in \mathbb{R}^F$, and $f_{ij} \in \mathbb{R}^{2F}$. The split information is used to compute the incremental node features as in Eq. (9).

$$\Delta x_i = W \left(\sum_{j \in \mathcal{N}(i)} \sigma \left(\Delta_{ij} + \xi_{ij} \left(f_{ij}, \bar{r}_{ij}, \mathcal{N}(ij) \right) \right) \right) + b \quad (9)$$

where $W \in \mathbb{R}^{F \times F}$, $b \in \mathbb{R}^F$, and ξ_{ij} represents the moment interaction function, as in Eq. (10).

$$\xi_{ij} = W^{\text{out}} \left(\sum_{ik \in \mathcal{N}(ij)} \sigma \left(W^{\text{in}} \left(m_{jik}^{(1)} \parallel m_{jik}^{(2)} \right) + b^{\text{in}} \right) \right) + b^{\text{out}} \quad (10)$$

where $W^{\text{in}} \in \mathbb{R}^{F \times 2F}$, $b^{\text{in}} \in \mathbb{R}^F$, $W^{\text{out}} \in \mathbb{R}^{F \times F}$, $b^{\text{out}} \in \mathbb{R}^F$, the symbol “ \parallel ” denotes the concatenation of two vectors. $\mathcal{N}(ij)$ represents the set of neighboring edges of the edge ij that share i as a common vertex for all triplets, as shown in the example in Fig. 1b, where $\mathcal{N}(ij) = \{ik, im\}$. Note that applying the non-linearity activation function before aggregation improve performance⁵².

In Eq. (10), the calculation of $m_{jik}^{(\nu)}$ (i.e., the message passed by triplet jik) involves the moments that are the core part of our algorithm, as shown in Fig. 1b, e, which are calculated as

$$m_{jik}^{(\nu)} = \sigma \left(W \left(f_{ij}^{(\nu)} \odot f_{ik}^{(\nu)} \odot M_{\theta_{jik}}^{(\nu)} \right) + b \right), \nu = 1, 2 \quad (11)$$

where $W \in \mathbb{R}^{F \times F}$, $b \in \mathbb{R}^F$, and “ \odot ” denotes element-wise product.

The $M_{\theta_{jik}}^{(\nu)}$ in Eq. (11) is the rank ν moment of the triplet jik , and is calculated as

$$M_{\theta_{jik}}^{(\nu)} = \langle \hat{r}_{ij}^{\otimes \nu}, \hat{r}_{ik}^{\otimes \nu} \rangle, \nu = 1, 2 \quad (12)$$

where $\hat{r}_{ij} = \frac{\vec{r}_{ij}}{r_{ij}}$ is the unit vector of \vec{r}_{ij} , $\hat{r}_{ij}^{\otimes \nu} = \underbrace{\hat{r}_{ij} \otimes \cdots \otimes \hat{r}_{ij}}_{\nu \text{ times}}$ is the Kronecker

product of ν copies of the vector $\hat{r}_{ij} \in \mathbb{R}^3$ and is a tensor of rank ν . Notation “ $\langle \cdot, \cdot \rangle$ ” signifies the contraction (product) of tensors, with $\nu = 1$ corresponding to the vector dot product and $\nu = 2$ corresponding to the Frobenius product of matrices. In other words, $M_{\theta_{jik}}^{(1)} = \langle \hat{r}_{ij}, \hat{r}_{ik} \rangle = \|\hat{r}_{ij}\| \times \|\hat{r}_{ik}\| \times \cos(\angle \hat{r}_{ij}, \hat{r}_{ik}) = \cos(\angle \hat{r}_{ij}, \hat{r}_{ik})$, and $M_{\theta_{jik}}^{(2)} = \langle \hat{r}_{ij}^{\otimes 2}, \hat{r}_{ik}^{\otimes 2} \rangle = \hat{r}_{ij}^{\otimes 2} : \hat{r}_{ik}^{\otimes 2} = \cos^2(\angle \hat{r}_{ij}, \hat{r}_{ik})$, reflecting the angular relationship within the triplet jik .

The $f_{ij}^{(\nu)}$ in Eq. (11) is calculated as

$$f_{ij}^{(1)}, f_{ij}^{(2)} = \text{Split} \left(\sigma \left(\text{MLP} \left(f_{ij} \right) \right) \right) \quad (13)$$

where f_{ij} is calculated using Eq. (8), and $f_{ij}^{(\nu)} \in \mathbb{R}^F$, $\nu = 1, 2$. They serve as the coefficients of moments of edge ij , which will be introduced in next subsection.

The rationale of moment interaction. We get motivation from the moment tensor descriptor⁴ for atom i , which is defined as a summation from edge moments as $\sum_{j \in \mathcal{N}(i)} f_{ij}^{(\nu)}(r_{ij}) \bar{r}_{ij}^{\otimes \nu}$, which encapsulated radial part $f_{ij}^{(\nu)}(r_{ij})$ and angular part $\bar{r}_{ij}^{\otimes \nu}$ containing angular information about the neighborhood $\mathcal{N}(i)$ and is tensor of rank ν ⁴. Our MGNN defines the moment of an edge ij as $f_{ij}^{(\nu)}(r_{ij}, z_i, z_j) \bar{r}_{ij}^{\otimes \nu}$ to account for both r_{ij} and the charges z_i and z_j , thereby enriching the function’s capacity to encapsulate the physicochemical context. Note that while $f_{ij}^{(\nu)}(r_{ij})$ is originally a scalar function of r_{ij} in ref. 4, in MGNN $f_{ij}^{(\nu)}(r_{ij}, z_i, z_j)$ is approximated using neural network.

As the edge moments are contracted to formulate triplet information, edge features (i.e., f_{ij} in Eqs. (8), (9), (11), and (13)) play the role of coefficients to multiply with triplet moment (i.e., $M_{ijk}^{(\nu)}$ in Eq. (12)) to construct the message passed by triplet (i.e., $m_{ijk}^{(\nu)}$ in Eq. (11)). Within a triplet (i.e., Fig. 1b) composed of nodes i, j, k and edges ij, ik , for $\nu = 1$, $\hat{r}_{ij}^{\otimes \nu} = \hat{r}_{ij}$, so that $M_{ijk}^{(1)} = \langle \hat{r}_{ij}, \hat{r}_{ik} \rangle = \cos(\angle \hat{r}_{ij}, \hat{r}_{ik})$, and for $\nu = 2$, $\hat{r}_{ij}^{\otimes \nu} = \hat{r}_{ij} \otimes \hat{r}_{ij}$, so that $M_{ijk}^{(2)} = \langle \hat{r}_{ij}^{\otimes 2}, \hat{r}_{ik}^{\otimes 2} \rangle = \cos^2(\angle \hat{r}_{ij}, \hat{r}_{ik})$ (i.e., the Frobenius product). The coefficients $f_{ij}^{(\nu)}$ and $f_{ik}^{(\nu)}$ for edges ij and ik are derived from Eq. (13).

As can be seen from Eq. (12), all the initial moment tensors are contracted to moment scalars (i.e., zero-order tensors) before the flow of message passing through triplets to edges to nodes, consuming less physical memory in computers. The outputs from message passing are also scalars. When the network is used to predict vectors or tensors, such as dipole moments or polarizabilities, these scalars and geometric information are then used as input formation in the output head to reconstruct vectors or tensors. These are fundamentally different from many existing equivariant models (i.e., GRACE⁵³, and CACE³⁹), where usually the higher-order tensor information is used throughout the entire information flow process.

Output head. When a scalar is needed, the last feedforward network transforms feature on each node into a scalar (Eq. (14)), as shown in Fig. 2a. We perform a sum aggregation over all nodes to predict scalar quantities like energy (Eq. (15)), as shown in Fig. 2b.

$$o_i = \text{MLP}(x_i^L) \quad (14)$$

$$E = \sum_i o_i \quad (15)$$

The force acting on atoms i , \vec{F}_i , is computed using auto-differentiation according to its definition as the negative gradient of the total energy with respect to the position of atom i :

$$\vec{F}_i(Z_1, \dots, Z_n, \vec{r}_1, \dots, \vec{r}_n) = -\frac{\partial E}{\partial \vec{r}_i}(Z_1, \dots, Z_n, \vec{r}_1, \dots, \vec{r}_n) \quad (16)$$

which gives an energy-conserving force field (Eq. (16)), as shown in Fig. 2b⁴⁴.

The molecular dipole moment $\vec{\mu}$ is the response of the molecular energy to an electric field $\nabla_{\vec{r}} E$ and, at the same time, the first moment of the electron density²⁶. It is often predicted using latent atomic charges, and in our model, we predict the dipole moment using Eq. (17) and the electronic spatial extent by Eq. (18), as shown in Fig. 2c, d respectively, consistent with PaiNN²⁶.

$$\vec{\mu} = \sum_i o_i (\vec{r}_i - \vec{c}) \quad (17)$$

$$\langle R^2 \rangle = \sum_i o_i (\|\vec{r}_i - \vec{c}\|)^2 \quad (18)$$

This assumes the center of mass at \vec{c} for brevity.

Specifically, when it is necessary to predict properties related to vectors or tensors, a gated equivariant block (for more information about rotational invariance and equivariance, please refer to Supporting Information S5)^{26,54}, as shown in Fig. 2f, is needed. If one needs to calculate the infrared and Raman spectra of a molecular system using deep learning potential functions, it is also necessary to compute the system's dipole moment and polarizability. In practical experiments, we utilize Eq. (17) to predict the system's dipole moment. However, to predict the molecular polarizability tensor, we require a vector feature to construct the tensor form. Therefore,

we first initialize the node's vector feature in the output head:

$$\vec{x}_i^0 = \sum_{j \in \mathcal{N}(i)} \sigma(Wx_j^L) \otimes \vec{r}_{ij} \quad (19)$$

where $W \in \mathbb{R}^{F \times F}$ is the learnable weight used to transform destination node feature to control how much the \vec{r}_{ij} contributes to the initial node vectorial features. $x_j^L \in \mathbb{R}^F$ is the destination node representation of the edge ij , and the superscript L denotes it is the output of the last interaction layer. $\vec{r}_{ij} \in \mathbb{R}^3$ is the vector in Cartesian coordinates from source node to destination node. Then through Eq. (19) vectorial features $x_i \in \mathbb{R}^{F \times 3}$ of atom i can be initialized, in which the superscript 0 denotes the initial vectorial features, as shown in Fig. 2e.

Considering the prediction of polarizability tensor (i.e., the response of the molecular dipole to an electric field²⁶), we construct polarizability tensors using

$$\alpha = \sum_{i=1}^N \alpha_0(x_i^L) I_3 + \vec{\nu}(\vec{x}_i) \otimes (\vec{r}_i - \vec{c}) + (\vec{r}_i - \vec{c}) \otimes \vec{\nu}(\vec{x}_i) \quad (20)$$

where I_3 is the identical matrix with shape 3×3 , and $\vec{\nu}(\vec{x}_i)$ transforms vector features (\vec{x}_i) initialized by Eq. (19) and passing them through two gated equivariant blocks^{26,54}, each yielding atom-wise scalars and vectors. The first term describes isotropic, atom-wise polarizabilities. The other two terms are the anisotropic components of the polarizability tensor. The atom positions subtracting the center of mass $\vec{r}_i - \vec{c}$ are used here to incorporate the global structure of the molecule²⁶. The calculation of polarizability tensor α is illustrated in Fig. 2f and the gated equivariant block in Fig. 2g.

We utilize the open-source deep learning frameworks PyTorch 2.0.1⁵⁰, SchNetPack 2.0⁵⁵, PyTorch Geometric 2.3.1⁵⁶, and PyTorch Lightning 2.1.3 for constructing and training MGNN (see more details in Supporting Information S3). Considering both moment orders $\nu = 1$ and $\nu = 2$, has better performance than considering $\nu = 1$ alone (details in Supporting Information S4).

Benchmark

We show that our MGNN architecture can achieve many state-of-the-art results on QM9 dataset^{41,42} and some state-of-the-art results on revised MD17 dataset⁴³, and MD17-ethanol⁴⁴. Furthermore, we validate the stability of MGNN on MD17-ethanol⁴⁴ and generalizability on 3BPA dataset⁴⁵ and computational efficiency on dataset 25-element high-entropy alloys⁵⁷. We refer the reader to the Supporting Information S3 for further training, data set and experimental details.

QM9. The QM9 dataset serves as a widely recognized benchmark for predicting various molecular properties at equilibrium^{41,42}. It comprises 130831 optimized structures with chemical elements (C, H, O, N, F) containing up to 9 heavy atoms (C, O, N, F). In our benchmarking of MGNN, we target 12 properties. To maintain consistency with prior research, MGNN is trained on 110000 examples, while 10000 molecules are reserved for validation purposes to regulate learning rate decay and enable early stopping. The remaining data is allocated as the test set, and results are averaged over three random splits. Table 1 presents a comparison with several state-of-the-art methods that learn the properties described above through a direct mapping from atomic coordinates and species. SchNet²⁸, Physnet³¹, Dimenet++⁴⁰, Equiformer³⁶, PaiNN²⁶, Allegro¹², SphereNet⁵⁸, TorchMD-Net⁵⁹, and TensorNet³⁸ are included in Table 1 for reference. MGNN attains state-of-the-art performance in seven target properties and delivers comparable results in the remaining targets.

Revised MD17. In our evaluation on the revised MD17 dataset⁴³, which contains 100,000 recomputed structures of 10 molecules from original MD17 dataset^{44,60,61}, we assess MGNN's capability to accurately learn the

Table 1 | MAE results on QM9 testing set for various chemical properties

Property	Unit	SchNet	Physnet	Dimenet++	Equiformer	PaiNN	Allegro	SphereNet	TorchMD-NET	TensorNet	MGNN
μ	D	0.033	0.053	0.03	0.011	0.012	-	0.026	0.011	-	0.01
α	a_0^3	0.235	0.062	0.044	0.046	0.045	-	0.046	0.059	-	0.041
ϵ_{HOMO}	meV	41	32.9	24.6	15	27.6	-	23	20	-	23.2
ϵ_{LUMO}	meV	34	24.7	19.5	14	20.4	-	18	18	-	17
$\Delta\epsilon$	meV	63	42.5	32.6	30	45.7	-	32	36	-	30^{a)}
$\langle R^2 \rangle$	a_0^2	0.073	0.765	0.331	0.251	0.066	-	0.292	0.033	-	0.04
ZPVE	meV	1.7	1.39	1.21	1.26	1.28	-	1.12	1.84	-	1.17
U_0	meV	14	8.15	6.32	6.59	5.85	4.7	6	6.15	4.3	4.1
U	meV	19	8.34	6.28	6.74	5.83	4.4	7	6.38	4.3	4.2
H	meV	14	8.42	6.53	6.63	5.98	4.4	6	6.16	4.3	4.1
G	meV	14	9.4	7.56	7.63	7.35	5.7	8	7.62	6	5.7
C_v	$\frac{cal}{mol K}$	0.033	0.028	0.023	0.023	0.024	-	0.021	0.026	-	0.023

Results for MGNN are averaged over three random splits. “-” denotes no results are reported in the referred papers for the corresponding properties. Best in bold.

^aNote that we predict $\Delta\epsilon$ simply by computing $\epsilon_{LUMO} - \epsilon_{HOMO}$, as this is precisely how it is calculated in by DFT calculations.

Table 2 | MAE results on revised MD17 dataset for energy and force predictions in units of [meV] and [meV Å⁻¹], respectively

Molecule		GAP	ACE	GemNet-(T/Q)	NequIP	Allegro	Tensor Net	BOTNet	MACE	MGNN
Aspirin	E	17.7	6.1	-	2.3	2.3	2.4	2.3	2.2	3.1
	F	44.9	17.9	9.5	8.2	7.3	8.9	8.5	6.6	9.1
Azobenzene	E	8.5	3.6	-	0.7	1.2	0.7	0.7	1.2	2.3
	F	24.5	10.9	-	2.9	2.6	3.1	3.3	3	6.2
Benzene	E	0.75	0.04	-	0.04	0.3	0.02	0.03	0.4	0.03
	F	6	0.5	0.5	0.3	0.2	0.3	0.3	0.3	0.2
Ethanol	E	3.5	1.2	-	0.4	0.4	0.5	0.4	0.4	0.3
	F	18.1	7.3	3.6	2.8	2.1	3.5	3.2	2.1	2.7
Malonaldehyde	E	4.8	1.7	-	0.8	0.6	0.8	0.8	0.8	0.8
	F	26.4	11.1	6.6	5.1	3.6	5.4	5.8	4.1	5.1
Naphthalene	E	3.8	0.9	-	0.2	0.5	0.2	0.2	0.5	0.5
	F	16.5	5.1	1.9	1.3	0.9	1.6	1.8	1.6	2.6
Paracetamol	E	8.5	4	-	1.4	1.5	1.3	0.3	1.3	2.4
	F	28.9	12.7	-	5.9	4.9	5.9	5.8	4.8	6.8
Salicylic acid	E	5.6	1.8	-	0.7	0.9	5.9	0.8	0.9	1.3
	F	24.7	9.3	5.3	4	2.9	4.6	4.3	3.1	6.4
Toluene	E	4	1.1	-	0.3	0.4	0.3	0.4	0.5	0.4
	F	17.8	6.5	2.2	1.6	1.8	1.7	1.9	1.5	2.4
Uracil	E	3	1.1	-	0.4	0.6	0.4	0.4	0.5	0.4
	F	17.6	6.6	3.8	3.1	1.8	3.1	3.2	2.1	2.7

Best in bold.

energies and forces of small molecules. This dataset comprises ten small, organic molecules, and we aim to predict both energy and forces on systems with fixed chemical composition exhibiting conformational changes. For each molecule, we allocate 950 configurations for training and 50 for validation, uniformly sampled from the entire dataset. Test error is evaluated on all remaining configurations in the dataset. A distinct model is trained for each trajectory. Table 2 presents a comparison with GAP², ACE⁵, GemNet-(T/D)²⁹, NequIP(l = 3)¹¹, Allegro¹², which are reported in ref. 12, and TensorNet 3L³⁸, BOTNet⁶², and MACE⁶³ which are reported in ref. 38. MGNN achieves state-of-the-art results for the molecules Ethanol, Benzene, and Uracil, and notably sets a new state-of-the-art for the energy MAE of Ethanol. Furthermore, in other molecular systems, MGNN surpasses previous descriptor-based machine learning

interatomic potentials (GAP, ACE), invariant GNNs(GemNet-(T/Q)) or achieves comparable results to other equivariant GNNs(NequIP(l = 3), Allegro, TensorNet 3 L, BOTNet, MACE).

MD17-ethanol. We also benchmark our MGNN in MD17-Ethanol Dataset⁴⁴, which is frequently used for stability test^{39,64}. Stability refers to the system’s ability to run sufficiently long trajectories while maintaining physically reasonable states. We followed the approach in ref. 39 and used the ethanol molecules in MD17 dataset for training and validation. The models compared include SchNet²⁸, DimeNet³⁰, sGDML⁶⁵, PaiNN²⁶, SphereNet³⁸, NewtonNet³³, GemNet-T²⁹, NequIP¹¹, and CACE³⁹. MGNN-X means that X samples are used to train MGNN. Except for CACE and MGNN-1000, all models were trained using 10,000

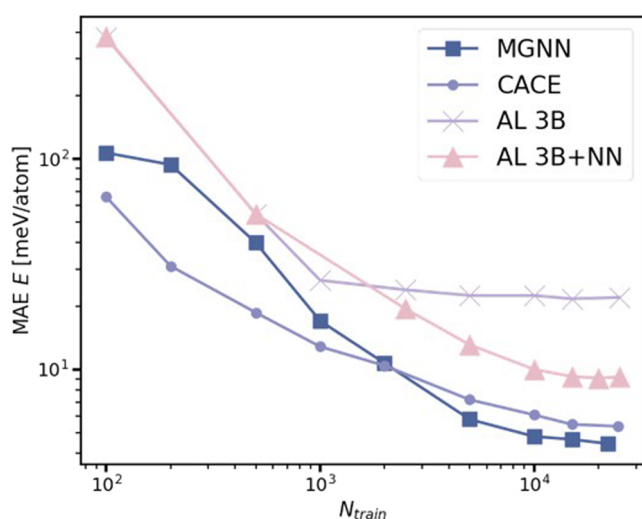
Table 3 | MAE of energy (E) and force (F) for ethanol in the MD-17 dataset (best in bold), in units of [meV] and [meV Å⁻¹], respectively

	SchNet	DimeNet	sGDML	PaiNN	SphereNet	NewtonNet	GemNet-T	NequIP	CACE	MGNN-1000	MGNN-10000
E	3.5	2.8	3	2.7	-	2.6	-	2.2	2.37	2.2	2.1
F	16.9	10	14.3	9.7	9	9.1	3.7	3.1	7	3.3	1.2
S†	247	26	-	86	33	-	169	300	300	300	300

The stability (S) in the unit of ps shows how long the simulation using the MLIP is physically realistic in MD of a maximum length of 300 ps at 500 K. Besides MGNN, the errors and the stability are obtained from ref. 39. The MGNN-1000 is trained on 1000 reference configurations and MGNN-10,000 is trained on 10,000 reference configurations.

Table 4 | Mean absolute error (MAE) of energy (E) and force (F) for the HEA25 dataset (best in bold), reported in units of [meV] and [meV Å⁻¹]

	AL 3B + NN	PET	CACE	MGNN
test, E	10	1.87	5.28	4.43
test, F	190	60.1	111	79

**Fig. 3 | Learning curve for different models.** Both alchemical learning (AL) models are from ref. 46, CACE is from ref. 39.

configurations. The stability (S) in Table 3 measures how long MD simulations at 500 K remain physically plausible (without pathological behavior or entering physically prohibitive states) in MD runs within 300 ps limit⁶⁴. The stability values are sourced from the CACE³⁹ literature. We find that MGNN potential can remain stable for the full 300 ps period, demonstrating its superior stability.

We also produce state-of-the-art results on force and energy prediction. Compared with CACE, MGNN-1000 achieves similar energy errors, but with smaller force errors. Compared with NequIP, MGNN-1000 achieves similar energy and force errors, yet MGNN-10000 achieves even smaller energy and force errors.

25-element high-entropy alloys. The HEA25 dataset from ref. 46 consists of 25,630 distorted crystalline structures containing 36 or 48 atoms on bcc or fcc lattices. We select a test set containing 1,000 configurations and use the remaining 24,630 data points for training and validation. The mean absolute errors (MAE) of energy and forces from MGNN for these tests are presented in Table 4. MGNN and PET⁵⁷ generalize significantly better than all other approaches. The learning curves from literature models³⁹ with respect to number of training samples are shown in Fig. 3, where MGNN is compared with other models, including a 3B model that has an atomic energy baseline and three-body terms, another 3B + NN model that further incorporates a

full set of pair potentials and a non-linear term built on top of contracted power spectrum features⁴⁶, and CACE. Overall, CACE and MGNN show superior performance. When the training data size is larger than 2,000, MGNN performs better than CACE.

3BPA. The 3BPA dataset was introduced in ref. 45 to evaluate the extrapolation capabilities towards other temperatures using machine-learned interatomic potentials. The training set consists of 500 configurations of the flexible drug-like molecule 3-(benzyloxy)pyridin-2-amine obtained from ab initio molecular dynamics at 300 K, while the test set includes a series of molecular dynamics simulations at 300 K, 600 K, and 1200 K. The root mean squared errors (RMSE) of energy and forces from MGNN for these tests are presented in Table 5, along with results from other state-of-the-art models. ACE^{(g)53} performs exceptionally well, achieving a significantly superior performance compared to other models. Models like MACE⁶³, NequIP¹¹, BOTNet⁶², and Allegro¹² yield similar results. CACE and MGNN exhibit comparable performance, with MGNN outperforming CACE at 300 K and 600 K, but underperforming at 1200 K. We attribute MGNN's poorer performance at 1200 K to small training data size (e.g., Fig. 3).

Computational cost. In Table 6, we select models from the Matbench-discovery library⁶⁶ (e.g., Equiformer v2³⁷, GemNet-T²⁹, PaiNN²⁶, SchNet²⁸, SCN⁶⁷) for comparison with our MGNN. The “F” following MGNN represents the number of scalar features, and “L” represents the number of layers. We compare training times and inference times using the HEA25 dataset. With the exception of MGNN, these models use the default parameters from the Matbench-discovery library⁶⁶. We select the first 1000 data points from HEA25 without any shuffling for training and inferencing for speed test. The chosen batch size is 2 to ensure all selected models could run smoothly on our NVIDIA GeForce RTX 3090 (24 GB) without errors.

As shown in Table 6, for models with the similar number of parameters, MGNN training and inference speeds are slightly slower than those of GemNet. This is due to the additional processing on angle information for preparing moments, and parameter-learning during the triplet information propagation, both of which slightly reduce computational efficiency. However, MGNN clearly outperforms Equiformer v2 in terms of speed on the same device. This is likely because Equiformer v2 requires the calculation of spherical harmonics by Clebsch-Gordan coefficients, which involves a larger computational load. MGNN's tensor computations are only performed in the first layer to produce contracted scalars, leading to higher computational efficiency.

Applications

We demonstrate MGNN's advantage as molecular potential for predicting molecular dynamics simulations of amorphous electrolytes, and rapid simulation of molecular spectra in vacuum conditions for organic molecules. We refer the reader to the Supporting Information S3 for further training, data set, experimental details, and molecular dynamics simulation settings.

Li-ion diffusion in a phosphate electrolyte. Our investigation into the MGNN's capabilities extends to the kinetic properties of Li-ion

Table 5 | Energy and Force RMSE for the 3BPA temperature transferability dataset (best in bold), reported in units of [meV] and [meV Å⁻¹]

Test set		ACE ^[9]	MACE	NequIP	BOTNet	Allegro	CACE	MGNN
300 K	E	1.7(0.1)	3.0 (0.5)	3.3 (0.1)	3.1 (0.1)	3.8 (0.1)	6.3	5.5 (0.3)
	F	7.6(0.2)	8.8 (0.3)	10.8 (0.2)	11.0 (0.1)	13.0 (0.2)	21.4	15.7 (0.3)
600 K	E	7.3 (0.8)	9.7 (0.5)	11.2 (0.1)	11.5 (0.6)	12.1 (0.5)	18	17.8 (0.9)
	F	19.2 (0.4)	21.8 (0.6)	26.4 (0.1)	26.7 (0.3)	29.2 (0.2)	45.2	39.6 (0.4)
1200 K	E	25.8 (1.6)	29.8 (1.0)	38.5 (1.6)	39.1 (1.1)	42.6 (1.5)	58	74.3 (2.7)
	F	61.7 (3.1)	62.0 (0.7)	76.2 (1.1)	81.5 (1.5)	83.0 (1.8)	113.8	142.6 (1.3)

Table 6 | Training time (ms) and inference time (ms) for energy and forces for single batch step for HEA25 dataset on an NVIDIA GeForce RTX 3090(24G)

Model (# Params)	equiformer_v2 31.1 M	GemNet - T 31.7 M	PailNN 51.8 M	SchNet 20.1 M	SCN 9.1 M	MGNN-F128-L3 1.9 M	MGNN-F256-L3 7.5 M	MGNN-F512-L3 30 M	MGNN-F512-L2 20.8 M
training time	186.24	44.7	334.59	19.78	15.67	23.81	48.64	121.83	82.01
inference time	84.17	22.78	126.99	10.19	7.23	11.9	21.62	48.72	33.46

Table 7 | Energy and Force MAE for the Li₃PO₄ and Ag datasets (best in bold), reported in units of [meV atom⁻¹] and [meV Å⁻¹]

Dataset		Allegro ¹²	MGNN
Li ₃ PO ₄	E	1.7	0.2
	F	73.4	22.9
Ag ^{a)}	E	0.397	0.168
	F	16.8	9.5

^{a)}To maintain consistency, we not only present the data for Li₃PO₄ in the table but also include another dataset demonstrated in their paper¹², namely, the Ag bulk crystal with a vacancy, simulated at 90% of the melting temperature. For further details, please refer to ref. 12. and our Supporting Information S3.

"E" denotes energy and "F" forces.

diffusion in Li₃PO₄ solid electrolyte, a material whose conductivity is intricately linked to its crystallinity. The Li₃PO₄ structure consists of 192 atoms. The simulations are conducted using a time step of 2 fs in the NVT ensemble, employing Nosé-Hoover thermostat⁶⁸. The dataset includes a 50 ps ab-initio molecular dynamic (AIMD) simulation in the molten liquid state as $T = 3000$ K, followed by a 50 ps AIMD simulation in the quenched state at $T = 600$ K. These trajectories are combined, and both the training set of 10,000 structures and the validation set of 1000 are randomly sampled from the combined dataset of 50,000 structures.

Note that the reference results for Allegro in Table 7 are sourced from their paper¹². The Ag model is trained and validated using 1000 structures, consist with ref. 12, resulting in a final training performance with energy mean absolute error (MAE) of 0.168 meV atom⁻¹ and force component MAE of 9.5 meV Å⁻¹ on a test set comprising 159 structures. For both the Li₃PO₄ and Ag datasets, we have achieved competitive results comparable to Allegro.

When evaluating the Li₃PO₄ test set for the quenched amorphous state, which the simulation is conducted on, we obtained a MAE in the energies of 0.2 meV atom⁻¹, along with a MAE in the force components of 22.92 meV Å⁻¹. Subsequently, we conduct a series of ten MD simulations starting from the initial structure of the quenched AIMD simulation, each lasting 50 ps at $T = 600$ K in the quenched state, to scrutinize how well MGNN captures the structure and kinetics compares 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 O-P-O (P central

atom). As depicted in Fig. 4a, b, MGNN adeptly reproduces both distribution functions. Concerning ion transport kinetics, we evaluate how accurately MGNN models the Li mean-square-displacement (MSD) in the quenched state. Once more, we observe excellent agreement with AIMD, as shown in Fig. 4c. The structure of Li₃PO₄ is depicted in Fig. 4d. Reference data for the Li₃PO₄ is obtained from ref. 12.

Molecular dynamics simulations are executed in LAMMPS⁶⁹ using the MGNN pair style, implemented in the SchNetPack 2.0⁵⁵ interface. Other details refer to Supporting Information S3.4. The RDF and ADF for Li₃PO₄ are computed with a maximum distance of 6 Å (RDF) and 2.5 Å (ADF). The simulation commences from the initial frame of the AIMD quenched simulation. RDF and ADF for MGNN are averaged over ten runs with different initial velocities, with the first 10 ps of the 50 ps simulation discarded in the RDF/ADF analysis to ensure equilibration is adequately accounted for.

Molecular spectra. MGNN is also employed to efficiently compute the infrared and Raman spectra of ethanol. Traditional methods, which might rely on the harmonic oscillator approximation from a singular molecular structure, often neglect critical phenomena such as the diversity of molecular conformations. To secure spectra of superior quality, it is imperative to undertake molecular dynamics simulations, which demand the prediction of forces at each timestep and the calculation of dipole moments and polarizabilities throughout the trajectory. The acquisition of infrared and Raman spectra through the Fourier transformation of time autocorrelation functions of dipole moments and polarizabilities respectively alleviates the complexity and computational intensity of using electronic structure methods. Given the necessity to account for nuclear quantum effects (NQE) for high-fidelity spectra, ring-polymer molecular dynamics (RPMD) simulations are employed, managing multiple replicas of the molecule simultaneously, thereby significantly increasing the computational burden²⁶.

A joint model has been trained on a dataset comprising 8000 conformations of ethanol, supplemented by an additional 1000 molecules each for validation and testing purposes. This model is tasked with the prediction of energies, forces, dipole moments, and polarizability tensors. Dipole moments and polarizability tensors are determined as described in Eqs. (17), (20), respectively. The unified model demonstrates precise predictions for energy, forces, dipole moments, and polarizabilities, as demonstrated in Table 8. Reference data for the ethanol molecule is sourced from ref. 27, and the experimental spectra recorded in the gas phase are obtained from refs. 70,71.

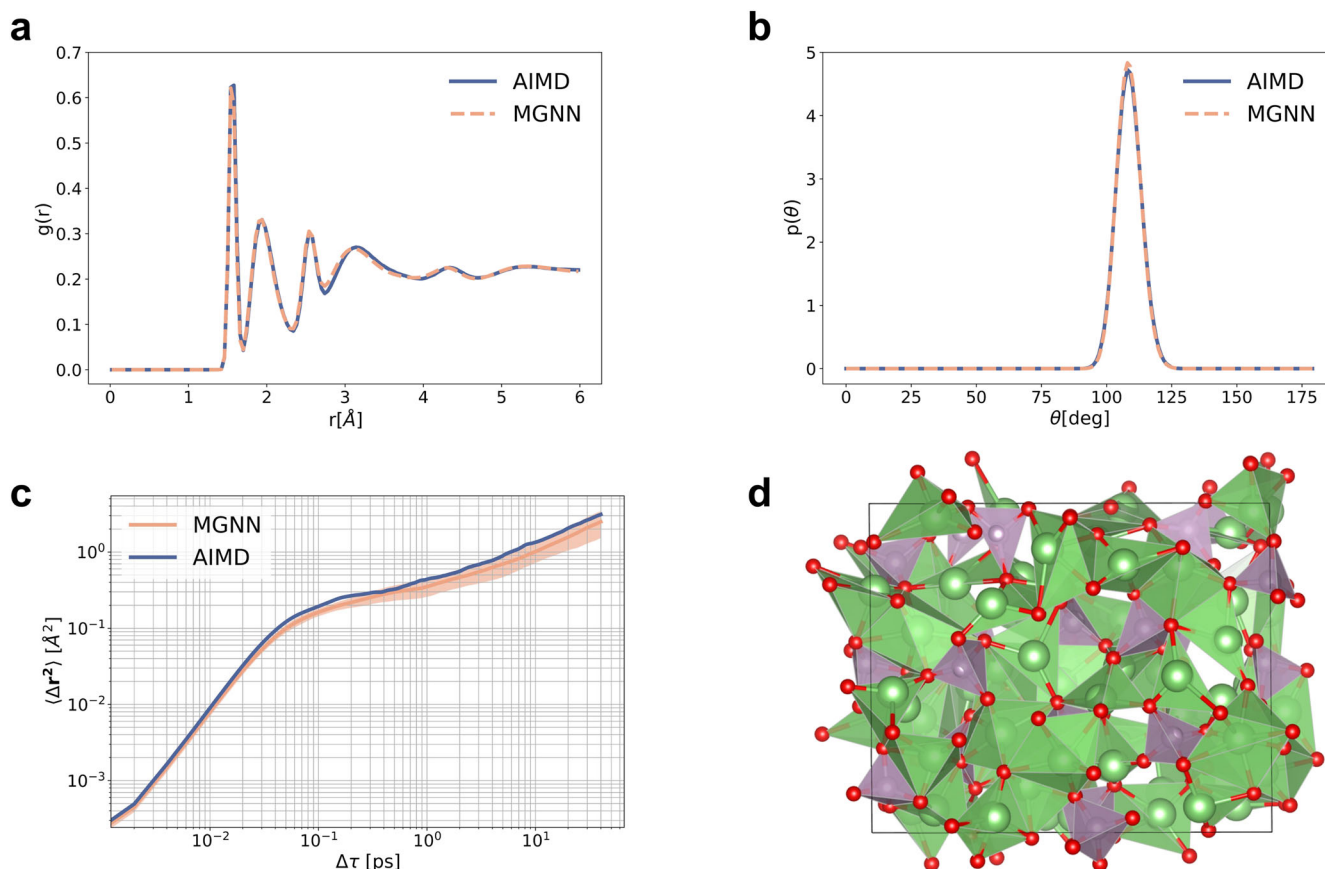


Fig. 4 | Structure properties of Li_3PO_4 and Li dynamic in Li_3PO_4 . a Radial distribution function. **b** Angular distribution function of tetrahedral bond angle. All defined probability density functions. **c** Comparison of the Li MSD of AIMD vs.

MGNN. Results are averaged over 10 runs of MGNN, shading indicates \pm one standard deviation. **d** The quenched Li_3PO_4 structure at $T = 600$ K. The VESTA⁷⁷ software was used to generate this figure.

Table 8 | Ethanol in vacuum results (best in bold)

Model	E [kcal mol ⁻¹]	F [kcal mol ⁻¹ Å ⁻¹]	$\bar{\mu}$ [D]	α [Bohr ³]
PaiNN ²⁶	0.027	0.150	0.003	0.009
FieldSchNet ²⁷	0.017	0.128	0.004	0.008
TensorNet ³⁸	0.008	0.058	0.003	0.007
MGNN	0.006	0.012	0.003	0.008

Mean absolute error for the prediction of energies (E), forces (F), dipole moments ($\bar{\mu}$), and polarizabilities (α), averaged over random 3 splits, with corresponding units between parentheses.

Results from Table 8 reveal that MGNN can acquire expressive atomic tensor embeddings, enabling simultaneous prediction of multiple molecular properties. Particularly noteworthy is MGNN's superior performance, with energy and force errors approximately a factor of two and three smaller, respectively, compared to FieldSchNet²⁷ and PaiNN²⁶. Additionally, MGNN achieves smaller MAE in force component prediction compared to the equivariant neural network TensorNet³⁸, while attaining results comparable to TensorNet for dipole moments and polarizabilities.

The MGNN potential is then used for classical MD and RPMD⁷². While both methods can calculate the spectra, they focus on different aspects. The peak positions and intensities of the ethanol infrared spectra (Fig. 5a) computed using classical MD and MGNN closely align with the static electronic structure reference, particularly evident in the C-H and O-H stretching regions at 3000 cm⁻¹ and 3900 cm⁻¹. This indicates MGNN's faithful reproduction of the original electronic structure method. However, in comparison to experimental data, both spectra exhibit shifts towards

higher frequencies. In contrast, the MGNN RPMD spectrum demonstrates excellent agreement with experimental results, highlighting the significance of NQEs. Similar trends are observed for the ethanol Raman spectrum (Fig. 5b), where the MGNN RPMD spectrum once again provides the most faithful reproduction of experimental observations.

More simulation details are shown in Supporting Information S3.6.

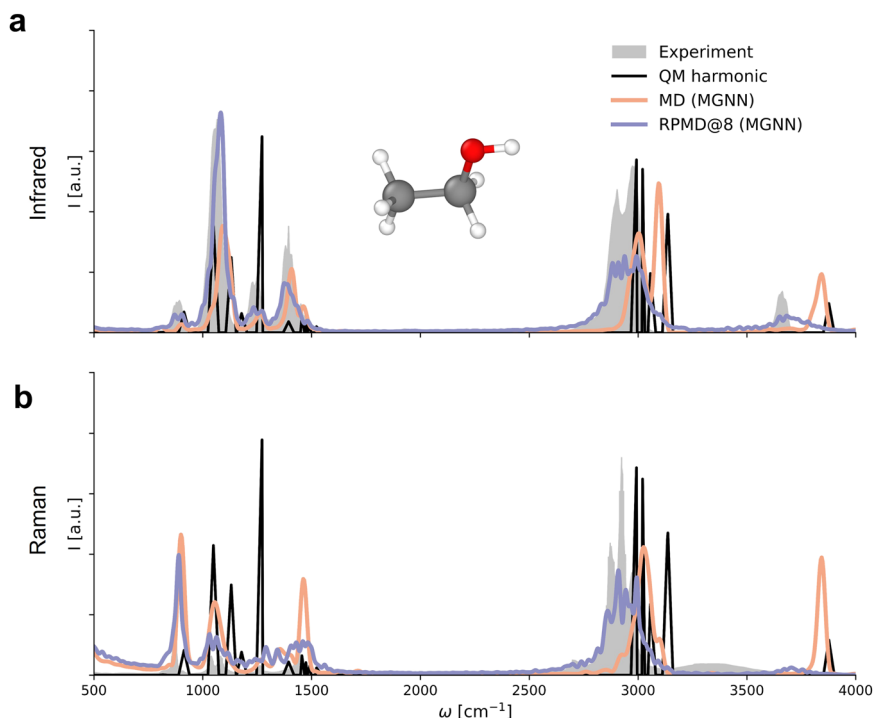
Discussion

In this work, we have introduced MGNN, a Graph Neural Network framework specifically designed for the rapid and accurate prediction of molecular properties. MGNN achieves multiple state-of-the-art results on the QM9 dataset and demonstrates comparable errors to the most recent state-of-the-art models on the revised MD17 dataset. On small molecule MD17-ethanol dataset for stability tests, MGNN shows lower energy errors, lower force errors, and highest stability value. On the 3BPA dataset, the generalization performance of MGNN is validated. On the HEA25 dataset, the learning and computational efficiency of MGNN is demonstrated. Finally, MGNN delivers remarkable consistency with ab-initio calculations, particularly in modeling the migration of lithium ions in electrolytes and in the calculation of infrared and Raman spectra of ethanol.

MGNN's output module is adept at interfacing with both vector and tensor modules, which enables the model to tailor its output representation to the property being predicted. While our approach does not incorporate vector computations within the molecular representation learning process, the inclusion of these computations in the final output module significantly enhances the model's computational efficiency. This is exemplified in our application of MGNN to predict vector and tensor features of molecules, such as the dipole moment and polarizability, which are crucial for

Fig. 5 | Infrared and Raman spectra of ethanol.

a Infrared spectra of ethanol. **b** Raman spectra of ethanol. Spectra calculated with the reference method using the harmonic oscillator approximations are shown in black (QM harmonic). Experimental results are shown in gray.



computing the infrared and Raman spectra of molecules like ethanol in vacuum conditions.

Looking ahead, the potential applications of MGNN are vast and varied, extending to areas such as molecular generation and wavefunction prediction. The rapid and accurate prediction of molecular properties is contingent upon highly precise reference methods, which often come with a high computational cost. As such, machine learning models like MGNN stand to play a pivotal role in surmounting these challenges, thereby driving forward the frontier of computational chemistry and materials science research.

Methods

QM9

The QM9 data set⁴² consists of 130,831 optimized structures of molecules that contain up to 9 heavy elements from C, N, O and F. On top of the structures, several quantum-chemical properties computed at the DFT B3LYP/6-31 G(2df,p) level of theory are provided.

We use the MGNN model with a cutoff of 5 Å, $F = 512$, $\beta = 8$, and 3 layers. More training details are provided in Supporting Information S3.2.

Revised MD17

The rMD17 data set⁴³ contains 100,000 recomputed structures of 10 molecules from MD17^{44,60,61}, a data set of small organic molecules obtained by running molecular dynamics simulations. The DFT PBE/def2-SVP level of theory, a very dense DFT integration grid and a very tight SCF convergence were used for the recomputations.

We use the MGNN model with a cutoff of 4.5 Å, $F = 512$, $\beta = 8$, and 2 layers. More training details are provided in Supporting Information S3.3.

Li₃PO₄

The Li₃PO₄ structure consists of 192 atoms. The Li₃PO₄ dataset utilized in this study comprises two segments: a 50 ps ab-initio molecular dynamic (AIMD) simulation in the molten liquid state as $T = 3000$ K, followed by a 50 ps AIMD simulation in the quenched state at $T = 600$ K. For further details, please refer to ref. 12.

We use the MGNN model with a cutoff of 4 Å, $F = 64$, $\beta = 8$, and 3 layers. More training and simulation details are provided in Supporting Information S3.4.

Ag

Ag system is created from a bulk face-centered-cubic structure with a vacancy, encompassing 71 atoms. For further details, please refer to ref. 12.

We use the MGNN model with a cutoff of 4.5 Å, $F = 128$, $\beta = 8$, and 3 layers. More training details are provided in Supporting Information S3.5.

Molecular spectra

Reference data for the ethanol molecule is sourced from ref. 27, and the experimental spectra recorded in the gas phase are obtained from refs. 70,71. The reference data for ethanol is generated by selecting 10,000 random configurations from the MD17 database⁴⁴, and re-evaluating them at the PBE0/def3-TZVP^{60,73} level of theory using the ORCA quantum chemistry package⁷⁴. SCF convergence is set to tight, and integration grid levels of 4 and 5 are employed during SCF iterations and the final computation of properties, respectively. Computations are expedited using the RIJK approximation⁷⁵.

We use the MGNN model with a cutoff of 4.5 Å, $F = 512$, $\beta = 8$, and 2 layers. More training and simulation details are provided in Supporting Information S3.6.

MD17-ethanol

MD17 is a collection of eight molecular dynamics simulations for small organic molecules. These data sets were introduced by ref. 44 for prediction of energy-conserving force fields using GDML. Each of these consists of a trajectory of a single molecule covering a large variety of conformations. To demonstrate the stability of MGNN on small organic molecules, we fit to ethanol structures from the MD17 database.

We use the MGNN model with a cutoff of 4.5 Å, $F = 512$, $\beta = 8$, and 2 layers. More training and simulation details are provided in Supporting Information S3.7.

25-element high-entropy alloys

The HEA dataset from ref. 46 consists of 25,630 distorted crystalline structures containing 36 or 48 atoms on bcc or fcc lattices. All the reference energies and forces are computed using density-functional theory (DFT), with the PBEsol exchange-correlation functional⁷⁶.

We use the MGNN model with a cutoff of 6 Å, $F = 128$, $\beta = 8$, and 3 layers. More training details are provided in Supporting Information S3.8.

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 Functional Theory with the ω B97X exchange-correlation functional and the 6-31 G(d) basis set. For details, we refer the reader to ref. 45.

We use the MGNN model with a cutoff of 4.5 Å, $F = 512$, $\beta = 8$, and 2 layers. More training details are provided in Supporting Information S3.9.

Data availability

The QM9 dataset was downloaded from https://figshare.com/collections/Quantum_chemistry_structures_and_properties_of_134_kilo_molecules/978904. Revised MD17 data can be obtained at https://figshare.com/articles/dataset/Revised_MD17_dataset_rMD17_/12672038. HEA25 dataset is from ref. 46, downloaded from <https://archive.materialscloud.org/record/2023.57>. Both Li_3PO_4 and Ag datasets are from ref. 12 and can be downloaded at <https://archive.materialscloud.org/record/2022.128>. Reference data for ethanol spectra is from ref. 27 and is available at <http://quantum-machine.org/datasets/>. MD17-ethanol is from <http://www.sgdml.org/#datasets>. The 3BPA dataset was downloaded from <https://pubs.acs.org/doi/full/10.1021/acs.jctc.1c00647>.

Code availability

The code for reproducing the results in this paper can be found at <https://github.com/JakechiC/MGNN>.

Received: 1 August 2024; Accepted: 30 January 2025;

Published online: 02 March 2025

References

- Behler, J. & Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **98**, 146401 (2007).
- 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).
- 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).
- Shapeev, A. V. Moment Tensor potentials: a class of systematically improvable interatomic potentials. *Multiscale Model. Simul.* **14**, 1153–1173 (2016).
- Drautz, R. Atomic cluster expansion for accurate and transferable interatomic potentials. *Phys. Rev. B* **99**, 014104 (2019).
- Zhang, L., Han, J., Wang, H., Car, R. & E, W. Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. *Phys. Rev. Lett.* **120**, 143001 (2018).
- 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. <https://doi.org/10.1109/SC41405.2020.00009> (IEEE, 2020).
- 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).
- Guo, Z. et al. Extending the limit of molecular dynamics with ab initio accuracy to 10 billion atoms. in *Proc. 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, 205–218. <https://doi.org/10.1145/3503221.3508425> (ACM, 2022).
- Nguyen-Cong, K. et al. Billion atom molecular dynamics simulations of carbon at extreme conditions and experimental time and length scales. In *Proc. International Conference for High Performance Computing, Networking, Storage and Analysis*, 1–12. <https://doi.org/10.1145/3458817.3487400> (ACM, 2021).
- Batzner, S. et al. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nat. Commun.* **13**, 2453 (2022).
- Musaelian, A. et al. Learning local equivariant representations for large-scale atomistic dynamics. *Nat. Commun.* **14**, 579 (2023).
- Rupp, M., Tkatchenko, A., Müller, K.-R. & Von Lilienfeld, O. A. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.* **108**, 058301 (2012).
- Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural Message Passing for Quantum Chemistry. In *Proc. 34th International Conference on Machine Learning*, 1263–1272 (PMLR, 2017).
- Cerioti, M. Beyond potentials: Integrated machine learning models for materials. *MRS Bull.* **47**, 1045–1053 (2022).
- Brockherde, F. et al. Bypassing the Kohn-Sham equations with machine learning. *Nat. Commun.* **8**, 872 (2017).
- Deng, B. et al. CHGNet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nat. Mach. Intell.* **5**, 1031–1041 (2023).
- Artrith, N., Morawietz, T. & Behler, J. High-dimensional neural-network potentials for multicomponent systems: Applications to zinc oxide. *Phys. Rev. B* **83**, 153101 (2011).
- López-Zorrilla, J. et al. ænet-PyTorch: a GPU-supported implementation for machine learning atomic potentials training. *J. Chem. Phys.* **158**, 164105 (2023).
- Zhang, L., Wang, H., Car, R. & E, W. Phase diagram of a deep potential water model. *Phys. Rev. Lett.* **126**, 236001 (2021).
- Chen, C. & Ong, S. P. A universal graph deep learning interatomic potential for the periodic table. *Nat. Comput. Sci.* **2**, 718–728 (2022).
- Choudhary, K. et al. Unified graph neural network force-field for the periodic table. *Digital Discov.* **2**, 346–355 (2023).
- Takamoto, S. et al. Towards universal neural network potential for material discovery applicable to arbitrary combination of 45 elements. *Nat. Commun.* **13**, 2991 (2022).
- Bereau, T., Andrienko, D. & von Lilienfeld, O. A. Transferable atomic multipole machine learning models for small organic molecules. *J. Chem. Theory Comput.* **11**, 3225–3233 (2015).
- Glielmo, A., Sollich, P. & De Vita, A. Accurate interatomic force fields via machine learning with covariant kernels. *Phys. Rev. B* **95**, 214302 (2017).
- Schütt, K., Unke, O. & Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *Proc. 38th International Conference on Machine Learning*, 9377–9388 (PMLR, 2021).
- Gastegger, M., Schütt, K. T. & Müller, K.-R. Machine learning of solvent effects on molecular spectra and reactions. *Chem. Sci.* **12**, 11473–11483 (2021).
- 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).
- Gasteiger, J., Becker, F. & Günnemann, S. Gemnet: Universal directional graph neural networks for molecules. *Adv. Neural Inf. Process. Syst.* **34**, 6790–6802 (2021).
- Gasteiger, J., Groß, J. & Günnemann, S. Directional message passing for molecular graphs. Preprint at <http://arxiv.org/abs/2003.03123> (2022).

31. 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).
32. Qiao, Z. et al. Informing geometric deep learning with electronic interactions to accelerate quantum chemistry. *Proc. Natl. Acad. Sci. U.S.A.* **119**, e2205221119 (2022).
33. Haghighatlari, M. et al. NewtonNet: a Newtonian message passing network for deep learning of interatomic potentials and forces. *Digital Discov.* **1**, 333–343 (2022).
34. Weiler, M., Geiger, M., Welling, M., Boomsma, W. & Cohen, T. S. 3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data. In *Advances in Neural Information Processing Systems*, vol. 31 (Curran Associates, Inc., 2018).
35. Thomas, N. et al. Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds. Preprint at <http://arxiv.org/abs/1802.08219> (2018).
36. Liao, Y.-L. & Smidt, T. Equiformer: equivariant graph attention transformer for 3d atomistic graphs. Preprint at <https://arxiv.org/abs/2206.11990> (2023).
37. Liao, Y.-L., Wood, B., Das, A. & Smidt, T. EquiformerV2: improved equivariant transformer for scaling to higher-degree representations. Preprint at <http://arxiv.org/abs/2306.12059> (2023).
38. Simeon, G. & De Fabritiis, G. TensorNet: cartesian tensor representations for efficient learning of molecular potentials. In *Proc. 37th International Conference on Neural Information Processing Systems*, 37334–37353. (Curran Associates Inc., Red Hook, NY, USA, 2023).
39. Cheng, B. Cartesian atomic cluster expansion for machine learning interatomic potentials. *npj Comput. Mater.* **10**, 1–10 (2024).
40. Gastegger, J., Giri, S., Margraf, J. T. & Günnemann, S. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. Preprint at <http://arxiv.org/abs/2011.14115> (2022).
41. Ruddigkeit, L., van Deursen, R., Blum, L. C. & Reymond, J.-L. Enumeration of 166 Billion organic small molecules in the chemical universe database GDB-17. *J. Chem. Inf. Model.* **52**, 2864–2875 (2012).
42. Ramakrishnan, R., Dral, P. O., Rupp, M. & Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **1**, 140022 (2014).
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. Kovács, D. P. et al. Linear atomic cluster expansion force fields for organic molecules: beyond RMSE. *J. Chem. Theory Comput.* **17**, 7696–7711 (2021).
46. Lopanitsyna, N., Fraux, G., Springer, M. A., De, S. & Ceriotti, M. Modeling high-entropy transition metal alloys with alchemical compression. *Phys. Rev. Materials* **7**, 045802 (2023).
47. Hendrycks, D. & Gimpel, K. Gaussian Error Linear Units (GELUs). Preprint at <http://arxiv.org/abs/1606.08415> (2023).
48. Ramachandran, P., Zoph, B. & Le, Q. V. Swish: a self-gated activation function. Preprint at <http://arxiv.org/abs/1710.05941> (2017).
49. Mishra, D. Mish: a self regularized non-monotonic activation function. Preprint at <http://arxiv.org/abs/1908.08681> (2020).
50. Paszke, A. et al. Pytorch: an imperative style, high-performance deep learning library. In *Adv. Neural Inf. Process. Syst.* **32**, 8026–8037 (2019).
51. He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778 (2016). <https://doi.org/10.1109/CVPR.2016.90>.
52. Brockschmidt, M. GNN-FiLM: Graph Neural Networks with Feature-wise Linear Modulation. In *Proc. 37th International Conference on Machine Learning*, 1144–1152 (PMLR, 2020).
53. Bochkarev, A., Lysogorskiy, Y. & Drautz, R. Graph atomic cluster expansion for semilocal interactions beyond equivariant message passing. *Phys. Rev. X* **14**, 021036 (2024).
54. Weiler, M., Hamprecht, F. A. & Storath, M. Learning steerable filters for rotation equivariant CNNs. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 849–858 <https://doi.org/10.1109/CVPR.2018.00095> (IEEE, Salt Lake City, UT, 2018).
55. Schütt, K. T., Hessmann, S. S. P., Gebauer, N. W. A., Lederer, J. & Gastegger, M. SchNetPack 2.0: a neural network toolbox for atomistic machine learning. *J. Chem. Phys.* **158**, 144801 (2023).
56. Fey, M. & Lenssen, J. E. Fast graph representation learning with PyTorch geometric. Preprint at <http://arxiv.org/abs/1903.02428> (2019).
57. Pozdnyakov, S. N. & Ceriotti, M. Smooth, exact rotational symmetrization for deep learning on point clouds. Preprint at <https://doi.org/10.48550/arXiv.2305.19302> (2023).
58. Liu, Y. et al. Spherical Message Passing For 3d Molecular Graphs. *Adv. Neural Inform. Process. Syst.* **36**, 79469–79501 (2022).
59. Thölke, P. & De Fabritiis, G. TorchMD-NET: equivariant transformers for neural network based molecular potentials. Preprint at <http://arxiv.org/abs/2202.02541> (2022).
60. 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).
61. 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).
62. Batatia, I. et al. The design space of E(3)-equivariant atom-centred interatomic potentials. *Nat. Mach. Intell.* **7**, 56–67 (2025).
63. Batatia, I., Kovács, D. P., Simm, G. N. C., Ortner, C. & Csányi, G. MACE: higher order equivariant message passing neural networks for fast and accurate force fields. *Adv. Neural Inform. Process. Syst.* **35**, 11423–11436 (2022).
64. Fu, X. et al. Forces are not enough: benchmark and critical evaluation for machine learning force fields with molecular simulations. Preprint at <http://arxiv.org/abs/2210.07237> (2023).
65. Chmiela, S., Sauceda, H. E., Poltavsky, I., Müller, K.-R. & Tkatchenko, A. sGDML: constructing accurate and data efficient molecular force fields using machine learning. *Comput. Phys. Commun.* **240**, 38–45 (2019).
66. Riebesell, J. et al. Matbench Discovery -- A framework to evaluate machine learning crystal stability predictions. Preprint at <http://arxiv.org/abs/2308.14920> (2024).
67. Zitnick, C. L. et al. Spherical channels for modeling atomic interactions. *Adv. Neural Inform. Process. Syst.* **35**, 8054–8067 (2022).
68. Martyna, G. J., Klein, M. L. & Tuckerman, M. Nosé–Hoover chains: the canonical ensemble via continuous dynamics. *J. Chem. Phys.* **97**, 2635–2643 (1992).
69. Thompson, A. P. et al. LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comput. Phys. Commun.* **271**, 108171 (2022).
70. Linstrom, P. J. & Mallard, W. G. The NIST Chemistry WebBook: a chemical data resource on the internet. *J. Chem. Eng. Data* **46**, 1059–1063 (2001).
71. Kiefer, J. Simultaneous acquisition of the polarized and depolarized raman signal with a single detector. *Anal. Chem.* **89**, 5725–5728 (2017).
72. Craig, I. R. & Manolopoulos, D. E. Quantum statistics and classical mechanics: real time correlation functions from ring polymer molecular dynamics. *J. Chem. Phys.* **121**, 3368–3373 (2004).
73. Weigend, F. & Ahlrichs, R. Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy. *Phys. Chem. Chem. Phys.* **7**, 3297 (2005).
74. Neese, F. The ORCA program system. *WIREs Comput. Mol. Sci.* **2**, 73–78 (2012).

75. Weigend, F. A fully direct RI-HF algorithm: Implementation, optimised auxiliary basis sets, demonstration of accuracy and efficiency. *Phys. Chem. Chem. Phys.* **4**, 4285–4291 (2002).
76. Csonka, G. I. et al. Assessing the performance of recent density functionals for bulk solids. *Phys. Rev. B* **79**, 155107 (2009).
77. Momma, K. & Izumi, F. VESTA: a three-dimensional visualization system for electronic and structural analysis. *J. Appl. Crystallogr.* **41**, 653–658 (2008).

Acknowledgements

This work was supported by National Key R&D Program of China (Grant Nos. 2022YFC3203900 and 2023YFC 2812700), the National Natural Science Foundation of China (no. 12002304, no. 12272337), the Distinguished Young Scientists Fund from the Natural Science Foundation of Zhejiang Province (no. LR23A020001), the “Pioneer” R&D Program of Zhejiang (Grant no. 2023C03007). The authors also acknowledge funding support from Zhejiang University, and Fundamental Research Funds for the Central Universities.

Author contributions

J.C. and S.Z. conceived the project, performed the calculations, analyzed the data and wrote 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/s41524-025-01541-5>.

Correspondence and requests for materials should be addressed to Shuze Zhu.

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-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, 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 licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence 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 licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025