npj | computational materials

Article

Published in partnership with the Shanghai Institute of Ceramics of the Chinese Academy of Sciences

https://doi.org/10.1038/s41524-025-01535-3

Efficient equivariant model for machine learning interatomic potentials

Ziduo Yang[1,2,7], Xian Wang[3,7], Yifan Li[1], Qiujie Lv[1,2], Calvin Yu-Chian Chen[4,5] ✉ & Lei Shen[1,6] ✉

In modern computational materials, machine learning has shown the capability to predict interatomic potentials, thereby supporting and accelerating conventional molecular dynamics (MD) simulations. However, existing models typically sacrifice either accuracy or efficiency. Moreover, efficient models are highly demanded for offering simulating systems on a considerably larger scale at reduced computational costs. Here, we introduce an efficient equivariant graph neural network (E^2GNN) that can enable accurate and efficient interatomic potential and force predictions for molecules and crystals. Rather than relying on higher-order representations, E^2GNN employs a scalar-vector dual representation to encode equivariant features. By learning geometric symmetry information, our model remains efficient while ensuring prediction accuracy and robustness through the equivariance. Our results show that E^2GNN consistently outperforms the prediction performance of the representative baselines and achieves significant efficiency across diverse datasets, which include catalysts, molecules, and organic isomers. Furthermore, we conduct MD simulations using the E^2GNN force field across solid, liquid, and gas systems. It is found that E^2GNN can achieve the accuracy of ab initio MD across all examined systems.

In the field of computational materials, calculating interatomic potentials is critical for obtaining energy and related physical quantities such as forces and atomic trajectories. Computational methods that use pre-fitted empirical functions to form interatomic potentials, such as classical molecular dynamics (MD), provide very fast but less accurate material-property calculations. Meanwhile, methods based on high-fidelity quantum-mechanics calculations, such as density functional theory (DFT) and ab initio molecular dynamics (AIMD), offer highly accurate energies and forces but require high computational costs.

To address the above dilemma, deep learning techniques, such as graph neural networks (GNNs), have been proposed for predicting interatomic potentials[1–4] or DFT Hamiltonian[5–8] in speed while preserving quantum mechanics-level accuracy. These models can be mainly divided into two categories: invariant models[9–15], such as CGCNN[16], SchNet[17], MEGNet[18], ALIGNN[19], and M3GNet[3], and equivariant models[20–24], such as MACE[1], NequIP[2], ScN[25], eScN[26], and DeepRelax[27]. To represent the molecule or crystal structure, GNNs typically use a graph where nodes are atoms and edges are bonds between atoms. Atomic interactions are then simulated by applying graph convolution operations on the graph, where an atom can access its neighboring atoms during this process (Fig. 1a).

Currently, the most widely used GNNs are designed with architectures that are invariant to transformations from the Euclidean group E(3), ensuring consistent output for energy relative to translations, rotations, and reflections. This is achieved by leveraging invariant features, such as bond lengths and angles, which remain constant under these transformations. Early models like CGCNN[16], SchNet[17], and MEGNet[18] primarily incorporate bond lengths, leading to challenges in distinguishing structures with identical bond lengths but different overall configurations (Fig. 1b). Later iterations, like DimeNet[28], ALIGNN[19], and M3GNet[3], improve upon this by integrating bond angles. Despite this improvement, they still struggled to differentiate between structures sharing the same angles (Fig. 1c). Recent models, such as GemNet[29] and SphereNet[30], propose considering dihedral angles in GNNs to unambiguously recognize the local structures (Fig. 1d). It is worth noting that distance and angular features are invariant representations which are only

[1]Department of Mechanical Engineering, National University of Singapore, Singapore, Singapore. [2]Artificial Intelligence Medical Research Center, School of Intelligent Systems Engineering, Shenzhen Campus of Sun Yat-sen University, Shenzhen, China. [3]Department of Physics, National University of Singapore, Singapore, Singapore. [4]AI for Science (AI4S)-Preferred Program, School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School, Shenzhen, China. [5]State Key Laboratory of Chemical Oncogenomics, School of Chemical Biology and Biotechnology, Peking University Shenzhen Graduate School, Shenzhen, China. [6]National University of Singapore (Chongqing) Research Institute, Chongqing, China. [7]These authors contributed equally: Ziduo Yang, Xian Wang. ✉e-mail: cy@pku.edu.cn; shenlei@nus.edu.sg

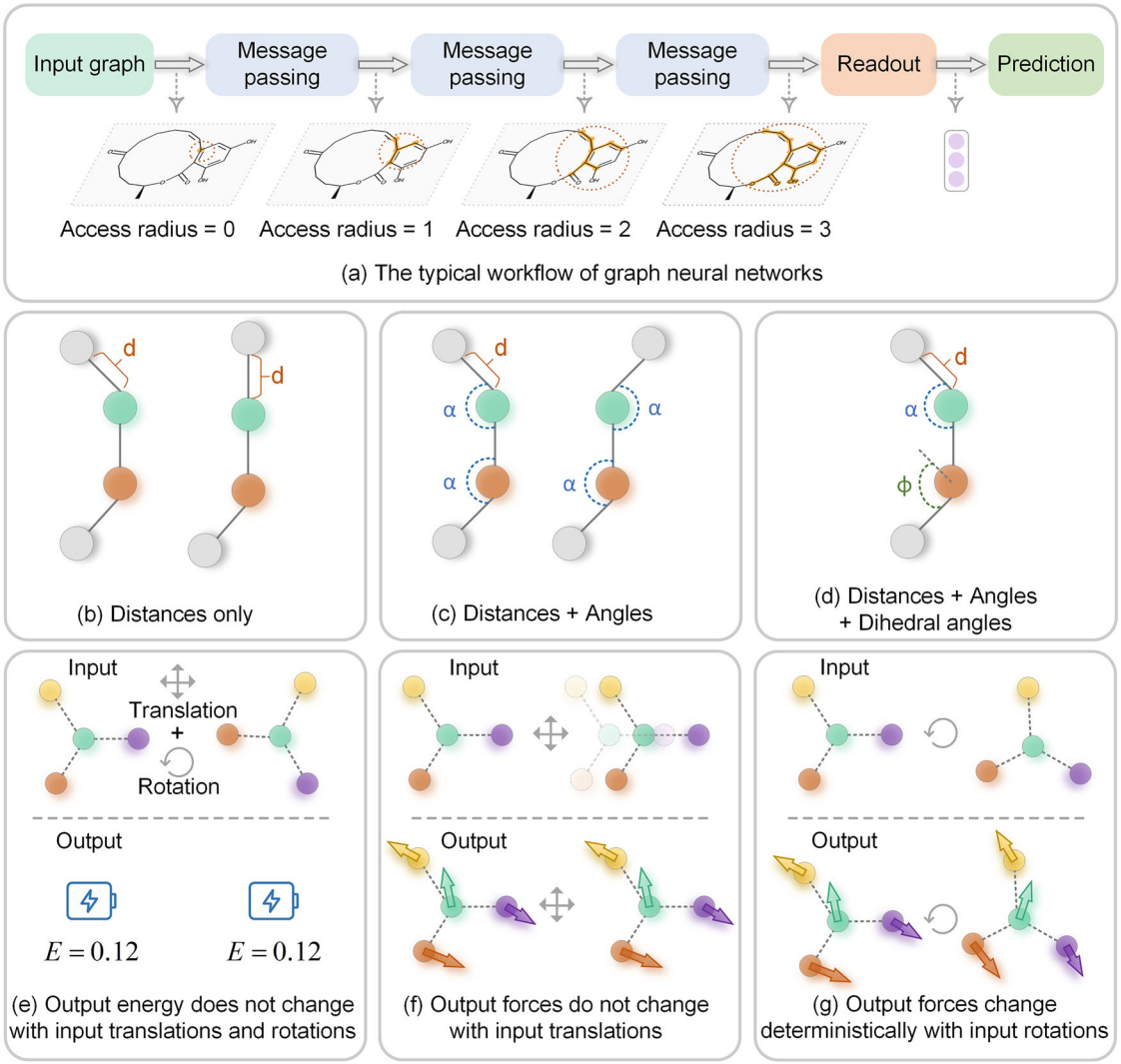

Fig. 1 | Overview of GNNs applied for material property prediction. a A typical GNN workflow is illustrated, emphasizing how GNNs can include a wider range of interactions (or message passing) by stacking multiple layers to extend the accessible radius. **b**, **c**, and **d** underscore the importance of integrating structural features into GNNs, which include: **b** distance only, **c** both distance and angles, and **d** all distance, angles, and dihedral angles. **e**, **f**, and **g** elucidate the concepts of invariance and equivariance within the context of energy and force prediction.

used to keep the geometric symmetry of crystals with respect to E(3) transformations rather than utilize the geometric symmetries in a more profound manner for increasing the prediction accuracy and the sample efficiency. Actually, the idea to leverage crystal symmetry for effectively describing the electron wave function and material properties was proposed a century ago. Felix Bloch demonstrated a translation-symmetry-based structural function to maintain the equivariance of wave functions in 1928. Such an equivariant idea in crystals with periodic structures in 3D space provides a powerful framework for accurately understanding the material properties and significantly reduces the cost of computation, opening a new era for computational material science. Figure 1e–g offer a concise elucidation of invariance and equivariance in the context of predicting energy and forces.

In this work, we integrate equivariance into GNNs for actively exploiting crystal symmetries and offering a richer geometrical representation compared to their invariant counterparts. Most importantly, our equivariant network is efficient, incorporating only scalar and vector features in a manner that preserves symmetry. Our efficient equivariant interaction graph neural network (E^2GNN) is different from previously equivariant GNNs which are based on high-order functions, such as the spherical harmonic function. While these high-order functions have been associated with increased accuracy compared to invariant GNNs, they typically come at the cost of increased computational expense[1,2,22–26,31–35]. For example, cutting-edge models like NequIP[2], MACE[1], ScN[25], and eScN[26] employ tensor product operations to combine input features and filters in an equivariant manner, which is computationally demanding in practice[25,26,36].

Our model aims to achieve both accurate and efficient predictions for interatomic potentials and forces. Specifically, we assign each node both scalars and vectors to represent equivariant features. E^2GNN combines these entities in a symmetry-preserving fashion to maintain equivariance. Equivariant E^2GNN surpasses scalar-only invariant models[16–18] in accuracy and generalization ability. It also offers an efficient structure compared to high-order tensor models[1,2,25]. We also conduct MD simulations using E^2GNN potential across solid, liquid, and gas systems. E^2GNN can achieve the ab initio MD accuracy with high computational efficiency across all examined systems, showing its accuracy and efficiency.

Methods

Network architecture

The proposed E^2GNN aims to enhance GNNs by incorporating equivariance, offering a richer geometric representation while retaining an efficient

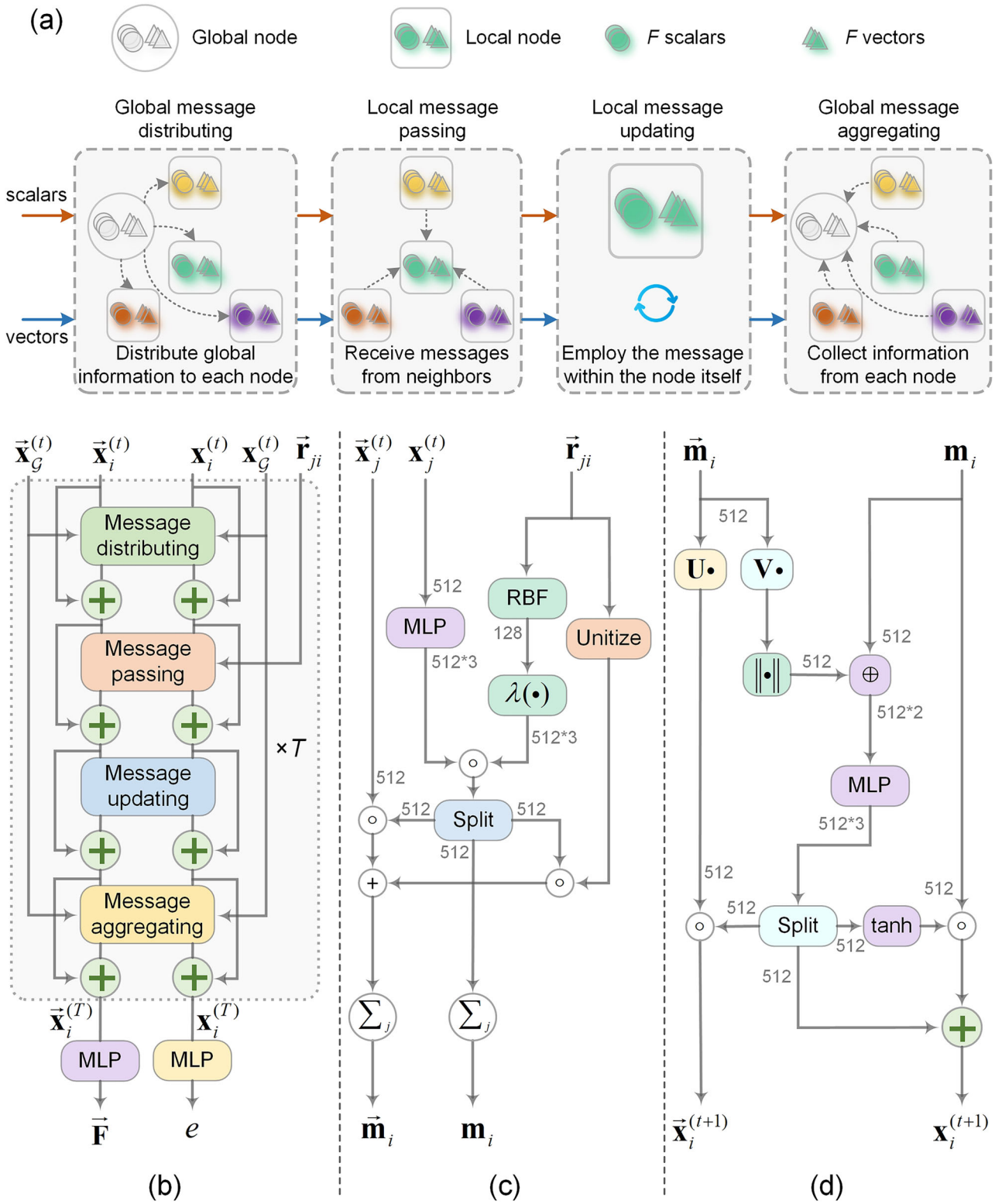

Fig. 2 | The overall architecture of E²GNN. a E²GNN uses message distributing, message passing, message updating, and message aggregating to iteratively update node representations. **b** E²GNN consists of T layers. **c** The message passing phase. **d** The message updating phase. The number of features after each operation is annotated in gray.

model. Each node in E²GNN is assigned scalars and vectors to represent invariant and equivariant features, respectively. This approach is similar to PaiNN[37] and NewtonNet[38], but with a more powerful message update scheme and global interaction modeling.

E²GNN gradually updates the node representations through four key processes: global message distributing, local message passing, local message updating, and global message aggregating, as illustrated in Fig. 2a and detailed in Fig. 2b–d. The local message-passing phase aggregates information from neighboring nodes to simulate two-body interactions. The local message updating phase integrates F scalars and F vectors within a node to update node representations. The global message aggregating and distributing phases promote

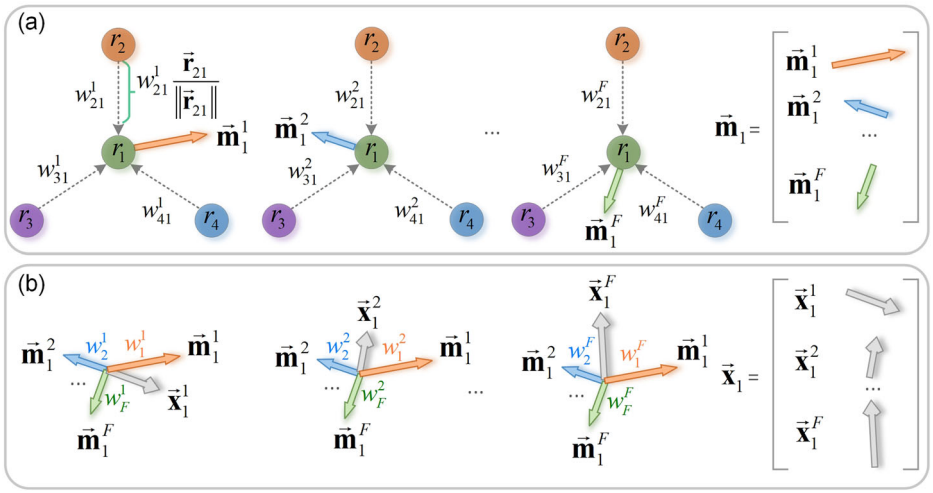

Fig. 3 | An illustration of how vector representation evolves during message passing phase and message updating phase. a An example explains how to calculate the intermediate vector representation $\vec{\mathbf{m}}_1$. For the sake of simplicity, we ignore the first term of Eqn. (2). The f-th vector in $\vec{\mathbf{m}}_1$ (i.e., $\vec{\mathbf{m}}_1^f$) can be interpreted as the force exerted by neighboring atom j on atom 1, where w_{j1}^f is force magnitude and $\frac{\vec{\mathbf{r}}_{j1}}{\|\vec{\mathbf{r}}_{j1}\|}$ is the force direction. Subsequently, the total force exerted on atom 1 is a linear combination of forces exerted on it by all other neighboring atoms j. Note that the forces are calculated F times in parallel. **b** The update to $\vec{\mathbf{x}}_1^f$ is achieved through a linear combination of F vectors within $\vec{\mathbf{m}}_1$. This computation is performed in parallel F times to obtain $\vec{\mathbf{x}}_1$.

long-range information exchange by maintaining a global scalar and vector.

Problem definition

In this work, the atomic structure is represented as a 3D interaction graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$, where $\mathcal{V}$ and $\mathcal{E}$ are sets of nodes and edges. $\mathcal{R}$ are sets of 3D coordinates of nodes. Each node is connected to its closest neighbors within a cutoff distance D with a maximum number of neighbors N, where D and N are predefined numbers. Each node $v_i \in \mathcal{V}$ has its scalar feature $\mathbf{x}_i \in \mathbb{R}^F$, vector feature $\vec{\mathbf{x}}_i \in \mathbb{R}^{F\times 3}$ (i.e., retaining F scalars and F vectors for each node), and 3D coordinate $\vec{\mathbf{r}}_i \in \mathbb{R}^3$. The scalar and vector features can be updated during training. We set the number of features F as a constant throughout the network. The scalar feature is initialized to an embedding only dependent on the atomic number as $\mathbf{x}_i^{(0)} = E(z_i) \in \mathbb{R}^F$, where z_i is the atomic number and E is an embedding layer that takes z_i as input and returns a F-dimensional feature. This embedding is similar to the one-hot vector but is trainable. The vector feature is set to $\vec{\mathbf{x}}_i^{(0)} = \vec{\mathbf{0}} \in \mathbb{R}^{F\times 3}$ at the initial step. We also define $\vec{\mathbf{r}}_{ij} = \vec{\mathbf{r}}_j - \vec{\mathbf{r}}_i$ as a vector from node v_i to node v_j. Norm $\|\cdot\|$ and dot product $\cdot$ are calculated along the spatial dimension, while concatenation $\oplus$ and Hadamard product $\circ$ are calculated along the feature dimension. Given a set of 3D interaction graphs, our goal is to learn a model f to predict the potentials and forces as $f(\mathcal{G}) = (e, \vec{\mathbf{F}})$, where $e \in \mathbb{R}^1$, $\vec{\mathbf{F}} \in \mathbb{R}^{M\times 3}$ and M is the number of nodes in $\mathcal{G}$.

Local message passing

In the t-th layer, during local message passing, a particular node v_i gathers messages from its neighboring scalar $\mathbf{x}_j^{(t)}$ and vector $\vec{\mathbf{x}}_j^{(t)}$, resulting in intermediate scalar and vector variables $\mathbf{m}_i$ and $\vec{\mathbf{m}}_i$ as follows:

$$\mathbf{m}_i = \sum_{v_j\in\mathcal{N}(v_i)} (\mathbf{W}_h\mathbf{x}_j^{(t)})\circ\lambda_h(\|\vec{\mathbf{r}}_{ji}\|) \tag{1}$$

$$\vec{\mathbf{m}}_i = \sum_{v_j\in\mathcal{N}(v_i)} (\mathbf{W}_u\mathbf{x}_j^{(t)})\circ\lambda_u(\|\vec{\mathbf{r}}_{ji}\|)\circ\vec{\mathbf{x}}_j^{(t)} + (\mathbf{W}_v\mathbf{x}_j^{(t)})\circ\lambda_v(\|\vec{\mathbf{r}}_{ji}\|)\circ\frac{\vec{\mathbf{r}}_{ji}}{\|\vec{\mathbf{r}}_{ji}\|} \tag{2}$$

Here, $\mathbf{W}_h, \mathbf{W}_u, \mathbf{W}_v \in \mathbb{R}^{F\times F}$ are learnable matrices. The functions λ_h, λ_u, and λ_v are the linear combination of Gaussian radial basis functions[17]. For the Eq. (2), the first term $(\mathbf{W}_u\mathbf{x}_j^{(t)})\circ\lambda_u(\|\vec{\mathbf{r}}_{ji}\|)\circ\vec{\mathbf{x}}_j^{(t)}$ propagates directional information $\vec{\mathbf{x}}_j^{(t)}$ obtained in the previous step to neighboring atoms, with $(\mathbf{W}_u\mathbf{x}_j^{(t)})\circ\lambda_u(\|\vec{\mathbf{r}}_{ji}\|)$ acting as a gate signal to control how many signals in the previous step can be preserved. We interpret the second term as the force exerted by atom j on atom i, where $(\mathbf{W}_v\mathbf{x}_j^{(t)})\circ\lambda_v(\|\vec{\mathbf{r}}_{ji}\|)$ is force magnitude and $\frac{\vec{\mathbf{r}}_{ji}}{\|\vec{\mathbf{r}}_{ji}\|}$ is the force direction. This is conceptually different from PaiNN's message function, which uses $\vec{\mathbf{r}}_{ij}$ instead of $\vec{\mathbf{r}}_{ji}$. Subsequently, $\sum_{v_j\in\mathcal{N}(v_i)}(\mathbf{W}_v\mathbf{x}_j^{(t)})\circ\lambda_v(\|\vec{\mathbf{r}}_{ji}\|)\circ\frac{\vec{\mathbf{r}}_{ji}}{\|\vec{\mathbf{r}}_{ji}\|}$ represents the total force exerted on atom i and is a linear combination of forces exerted on it by all other atoms. Figure 3a shows an example of how the F forces are calculated.

Local message updating

The local message updating phase aims to aggregate F scalars and vectors within $\mathbf{m}_i$ and $\vec{\mathbf{m}}_i$, respectively, to obtain new scalar $\mathbf{x}_i^{(t+1)}$ and new vector $\vec{\mathbf{x}}_i^{(t+1)}$, as shown in Fig. 3b. Specifically, the scalar representation $\mathbf{x}_i^{(t+1)}$ and vector representation $\vec{\mathbf{x}}_i^{(t+1)}$ are updated according to the following equations:

$$\mathbf{x}_i^{(t+1)} = \mathbf{W}_s(\mathbf{m}_i\oplus\|\mathbf{V}\vec{\mathbf{m}}_i\|) + \tanh(\mathbf{W}_g(\mathbf{m}_i\oplus\|\mathbf{V}\vec{\mathbf{m}}_i\|))\mathbf{m}_i \tag{3}$$

$$\vec{\mathbf{x}}_i^{(t+1)} = (\mathbf{W}_h(\mathbf{m}_i\oplus\|\mathbf{V}\vec{\mathbf{m}}_i\|))\circ(\mathbf{U}\vec{\mathbf{m}}_i) \tag{4}$$

where $\oplus$ denotes concatenation, $\mathbf{W}_s, \mathbf{W}_g, \mathbf{W}_h \in \mathbb{R}^{F\times 2F}$, and $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{F\times F}$. The term $\tanh(\mathbf{W}_g(\mathbf{m}_i\oplus\|\mathbf{V}\vec{\mathbf{m}}_i\|))$ acts as a gate controlling how much invariant signal from the previous layer is preserved. This design allows for selective updating of node representations, which can help in preserving important features from the previous layers while incorporating new information.

Global message distributing and aggregating

In many GNN architectures, information primarily flows locally between directly connected nodes. While increasing the depth of GNNs can broaden receptive fields, it can also lead to optimization instabilities, such as vanishing gradients and representation oversmoothing. To facilitate a more effective global communication channel across the entire graph, we propose a global message distributing and aggregating scheme by creating a global scalar $\mathbf{x}_\mathcal{G}$ and a global vector $\vec{\mathbf{x}}_\mathcal{G}$. Specifically, we initialize the global scalar and vector as $\mathbf{x}_\mathcal{G}^{(0)} \in \mathbb{R}^F$ and $\vec{\mathbf{x}}_\mathcal{G}^{(0)} = \vec{\mathbf{0}} \in \mathbb{R}^{F\times 3}$, where $\mathbf{x}_\mathcal{G}^{(0)}$ is trainable. The global message distributing operates before the local message passing, distributing the global scalar and vector at the current step to each node

using the following equations:

$$\mathbf{x}_i^{(t)} = \phi(\mathbf{x}_i^{(t-1)} \oplus \mathbf{x}_{\mathcal{G}}^{(t-1)}) + \mathbf{x}_i^{(t-1)} \tag{5}$$

$$\vec{\mathbf{x}}_i^{(t)} = \mathbf{W}(\vec{\mathbf{x}}_i^{(t-1)} + \vec{\mathbf{x}}_{\mathcal{G}}^{(t-1)}) + \vec{\mathbf{x}}_i^{(t-1)} \tag{6}$$

where $\phi : \mathbb{R}^{2F} \to \mathbb{R}^F$ refers to a multi-layer perceptron layer (MLP), and $\mathbf{W} \in \mathbb{R}^{F\times F}$ is a trainable matrix. After local message updating, the global scalar and vector are updated using the node representations at the current step with the following equations:

$$\mathbf{x}_{\mathcal{G}}^{(t+1)} = \phi\left(\left(\frac{1}{|\mathcal{G}|}\sum_{v_i\in\mathcal{G}}\mathbf{x}_i^{(t)}\right) \oplus \mathbf{x}_{\mathcal{G}}^{(t)}\right) + \mathbf{x}_{\mathcal{G}}^{(t)} \tag{7}$$

$$\vec{\mathbf{x}}_{\mathcal{G}}^{(t+1)} = \mathbf{W}\left(\left(\frac{1}{|\mathcal{G}|}\sum_{v_i\in\mathcal{G}}\vec{\mathbf{x}}_i^{(t)}\right) + \vec{\mathbf{x}}_{\mathcal{G}}^{(t)}\right) + \vec{\mathbf{x}}_{\mathcal{G}}^{(t)} \tag{8}$$

E^{2}GNN is strictly equivariant to rotation and translation, as proven in SM Section 1. Its design also guarantees the extensivity of energy, i.e., the predicted energy of E^{2}GNN is linear with respect to the simulation box size, as proven in SM Section 2.

Predicting potentials and forces

To predict the potential e, we utilize an MLP layer $\phi : \mathbb{R}^F \to \mathbb{R}^1$. This layer learns atom-wise potentials $e_i \in \mathbb{R}^1$ from the scalar representation $\mathbf{x}_i^{(T)}$, which is obtained at the last graph convolution layer (referred to as the T-th layer). The total potential is then calculated as the sum of the atom-wise potentials:

$$e = \sum_{v_i\in\mathcal{G}} e_i \tag{9}$$

The forces are predicted using the vector representation $\vec{\mathbf{x}}_i^{(T)}$ as $\vec{\mathbf{F}} = \mathbf{W}_f\vec{\mathbf{x}}_i^{(T)}$, where $\mathbf{W}_f \in \mathbb{R}^{1\times F}$. This approach offers the advantage of creating a efficient model by directly computing forces, which bypasses the computationally expensive tasks of calculating the gradient of the potential energy (first derivative) and the second derivative during backpropagation for model parameter updates. Although the directly predicted forces are not energy conserving, they can still facilitate MD simulations, especially when a thermostat is employed to regulate the temperature[36,39].

Implementation details

The E^{2}GNN model is implemented using PyTorch. Experiments are conducted on an NVIDIA GeForce RTX A4000 with 16 GB of memory. The training objective aims to minimize the loss function defined as:

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N}\left(\alpha|e_n - e_n^l| + \beta\frac{1}{3M}\sum_{m=1}^{M}\sum_{k=1}^{3}|\vec{\mathbf{F}}_{nmk} - \vec{\mathbf{F}}^{\,l}_{nmk}|\right) \tag{10}$$

where e_n^l represents the ground truth energy of n-th sample, $\vec{\mathbf{F}}^{\,l}_{nmk}$ is the ground truth force of k-th dimension of m-th atom in n-th sample. The variables N and M denote the sample size and the number of atoms in each sample, and α and β denote the weights assigned to the energy and force losses, respectively.

Results

Model performance

We first evaluate E^{2}GNN in the structure to energy and forces (S2EF) task using Open Catalyst 2020 (OC20) dataset[40]. The purpose of this task is to predict energies and forces corresponding to each trajectory during structural relaxation. The OC20 dataset encompasses 1,281,040 density functional theory (DFT) relaxations with 264,890,000 single-point calculations and spans a vast range of materials, surfaces, and adsorbates. It has been reported that training a GNN model on the whole OC20 dataset requires hundreds or even thousands of days[26]. Therefore, we only use a subset of it: OC20-200 K (N = 200, 000). The dataset is split into a training set and an internal validation set with a ratio of 8:2, where the internal validation set is used to select the best model for testing. Finally, the select models are tested on four external validation sets provided by the OC20 project: in Domain (ID), out-of-domain adsorbate (OOD Ads.), out-of-domain catalyst (OOD Cat.), and OOD Both (both the adsorbate and catalyst are not seen in the training set). Each external validation set contains approximately 1M data points. We compare E^{2}GNN with seven representative baseline models: CGCNN[16], SchNet[17], MACE[1], PaiNN[37], PaiNN_Direct[41], DimeNet++[42], and GemNet-dT[29]. PaiNN_Direct is a modified version of PaiNN designed for making direct force predictions. The hyperparameter configuration for each model is provided in SM Section 3. All models share the same training, internal validation, and external validation sets, and are trained to predict adsorption energy and per-atom forces simultaneously. The performance of the models is evaluated based on the mean absolute error (MAE).

From Table 1, we have two key observations. First, E^{2}GNN achieves performance close to GemNet-dT and outperforms other baseline models. Second, PaiNN_Direct surpasses PaiNN, which aligns with previously reported results[43]. Additional experimental results on a smaller subset, OC20-50 K, are available in SM Section 4.

Next, we compare the computational costs among E^{2}GNN and five accurate models: MACE, PaiNN, PaiNN_Direct, DimeNet++, and GemNet-dT. Figure 4 illustrates the model complexity in terms of training time, inference time, the number of parameters, and floating-point operations (FLOPs). It is worth noting that the number of parameters and FLOPs are estimated using PyTorch-OpCounter, which fails when applied to

Table 1 | Comparison results of the proposed E^{2}GNN and baselines on S2EF task of four external validation sets of OC20 in terms of energy MAE (meV) and forces MAE (meV/Å), where all models are trained on OC20-200K

Model	ID		OOD Ads.		OOD Cat.		OOD Both	
	Energy	Forces	Energy	Forces	Energy	Forces	Energy	Forces
CGCNN	1111	75.0	1261	80.7	1097	74.1	1383	91.9
SchNet	975	59.6	1077	66.8	975	59.5	1204	77.9
MACE	565	51.3	657	59.8	589	51.3	802	68.6
PaiNN	482	52.7	570	58.8	499	52.4	704	74.0
PaiNN_Direct	457	46.6	571	53.5	497	46.9	689	63.4
DimeNet++	497	48.7	547	55.9	522	48.6	671	65.3
GemNet-dT	443	**41.3**	516	**46.7**	548	**41.6**	717	**55.3**
E^{2}GNN	**415**	43.9	**502**	49.6	**452**	43.9	**613**	58.1

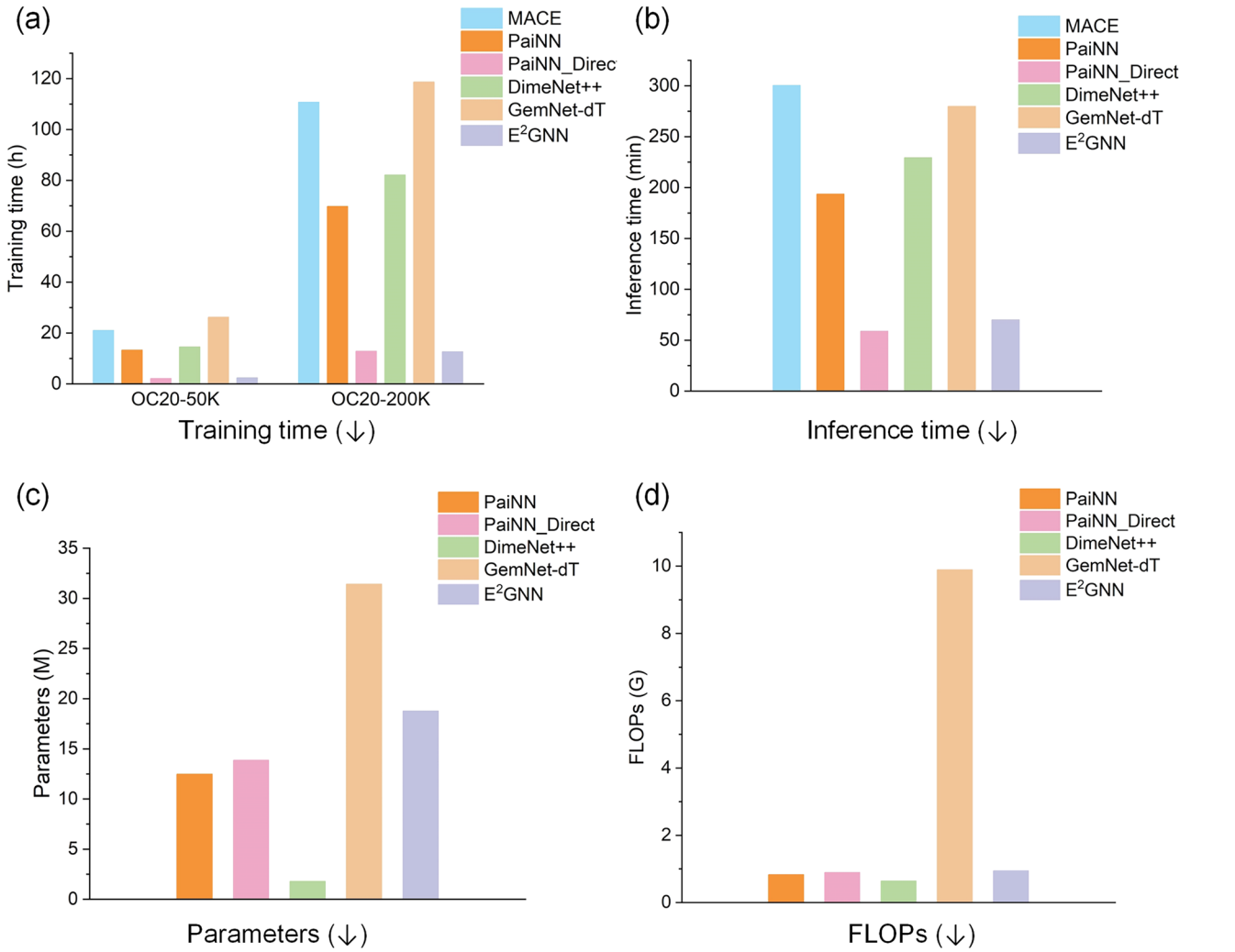

Fig. 4 | Computational costs of the representative models (the lower the better for all metrics). a Training time. **b** Mean inference time. **c** Number of parameters. **d** FLOPs.

Fig. 5 | Pareto front plot comparing ML models for force accuracy and inference time on the OC20-200 K dataset. The x-axis represents inference time (min), while the y-axis shows force MAE (meV/Å). Models on the Pareto front, including our proposed model, demonstrate an optimal balance between computational efficiency and prediction accuracy.

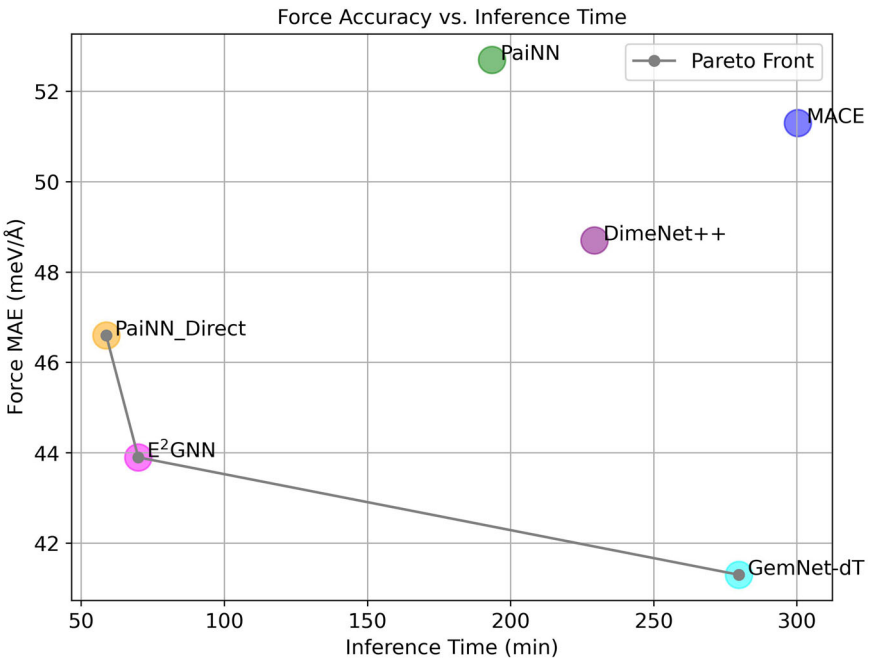

Table 2 | Ablation study on the OC20-50 K and OC20-200K datasets to demonstrate the effectiveness of E²GNN

Dataset	Models	ID		OOD Ads.		OOD Cat.		OOD Both	
		Energy	Forces	Energy	Forces	Energy	Forces	Energy	Forces
OC20-50K	Vanilla	568	53.1	691	57.9	602	52.6	847	68.3
	Vanilla + NMU	552	52.4	634	57.4	615	51.8	785	67.7
	E²GNN	**531**	**51.2**	**626**	**56.2**	**555**	**50.7**	**750**	**66.0**
	Improvement	6.52%	3.58%	9.40%	2.93%	7.81%	3.61%	11.45%	3.37%
OC20-200K	Vanilla	434	44.7	519	50.7	487	45.0	659	59.7
	Vanilla + NMU	427	44.7	525	50.6	480	44.8	662	59.4
	E²GNN	**415**	**43.9**	**502**	**49.6**	**452**	**43.9**	**613**	**58.1**
	Improvement	4.38%	1.79%	3.28%	2.17%	7.18%	2.44%	6.98%	2.68%

Performance is evaluated in terms of energy MAE (meV) and forces MAE (meV/Å). The improvement is calculated by comparing E²GNN with the Vanilla model.

MACE. We have three key observations. First, E²GNN offers approximately ten times faster training and inference speeds than GemNet-dT and MACE, and about five times faster than DimeNet++ and PaiNN. Second, E²GNN has a similar computational cost to PaiNN_Direct but demonstrates better performance. Third, the computational cost of PaiNN_Direct is significantly lower than that of PaiNN. Considering that the main difference between PaiNN_Direct and PaiNN is how they compute forces, we can conclude that computing forces by taking the derivative of the energy with respect to positions is more computationally demanding.

Figure 5 shows a Pareto front plot, with the two axes representing inference time and force MAE. The Pareto front highlights that our model achieves one of the best trade-offs between speed and accuracy among the benchmarked ML models. Overall, our results show that direct force prediction improves both efficiency and performance, consistent with previous studies[15,32,43]. Besides the OC20 database, we compare E²GNN with other models on MD17 and ISO17 databases. Our E²GNN model also surpasses those benchmark models on these two databases (see SM Sections 7 and 8), showing the universality of E²GNN.

Ablation study

The novelty of E²GNN lies in two aspects: a novel message updating (NMU) scheme and the global message aggregating and distributing phases to promote long-range information exchange among atoms. To demonstrate the effectiveness of these strategies, we compare E²GNN with two baseline models: Vanilla and Vanilla + NMU. The Vanilla model excludes both the NMU scheme and the global message phases, using PaiNN's message updating instead. The Vanilla + NMU model includes only the NMU scheme. As shown in Table 2, E²GNN notably outperforms both baseline models. Besides, it is worth noting that such two strategies are carried out on nodes rather than edges. This design choice offers a computational advantage because the number of nodes is typically significantly smaller than the number of edges in a graph. Note that the Vanilla model differs from PaiNN_Direct in several ways. Additional details are provided in SM Section 3.

Molecular dynamics simulations

Molecular dynamics simulations provide vital atomistic insights, but the accuracy versus efficiency trade-off has long been a challenge. Classical MD, which relies on empirical interatomic potentials, is computationally efficient but sacrifices accuracy. In contrast, AIMD, which integrates first-principles methods such as density functional theory, provides higher accuracy but at a high computational cost. Recently, machine learning-based force fields have emerged as a promising solution to enhance the speed of MD simulations by several orders of magnitude while maintaining quantum chemical accuracy.

To validate the interatomic potentials of our E²GNN model, we demonstrate that E²GNN performs competitively against established baseline models such as DeepPot-SE[44], SchNet[17], DimeNet[28], ForceNet[15], GemNet-T[29], GemNet-dT[29], and NequIP[2]. We evaluate the models' performance using the mean absolute error (MAE) of the radial distribution function (RDF), stability, and diffusivity. Our definitions of stability and diffusivity align with those used in MDSim[36]. A simulation is considered "unstable" if deviations exceed predefined thresholds, leading to the sampling of highly nonphysical structures. Diffusivity measures the time-correlation of translational displacement. For detailed experimental settings and metrics, refer to SM Sections 5 and 6.

We consider three systems: LiPS (solid), H_2O (liquid), and CH_4 (gas), as depicted in Fig. 6a, where the data for H_2O and CH_4 (gas) are collected in-house to assess E²GNN's effectiveness in liquid and gaseous systems. We conduct a 50-picosecond simulation for each system, starting from a randomly selected test configuration. The training, validation, and testing sets for the LiPS system are consistent with those used in MDSim. The results for the LiPS system are summarized in Table 3. Although the forces predicted by E²GNN are not from the widely used energy-conserving method, the MD simulation performance remains competitive with that of energy-conserving models. Figure 6b, c shows the $h(r)$ and RDF for AIMD and E²GNN across the three systems, revealing their strong alignment. Using the same stability criterion as for the LiPS system, E²GNN also demonstrates consistent stability throughout the 50 ps simulations in the H_2O (liquid) and CH_4 (gas) systems. These results highlight E²GNN's potential as a crucial tool in computational materials science.

Since the introduction of global entities can potentially lead to inaccuracies in scenarios where distant parts of the system are improperly influenced, we further tested E²GNN on a larger system consisting of 2241 atoms. The results presented in SM Fig. S2 show that the RDF remains consistent between the supercell ML trajectory and the DFT trajectory. This suggests that long-range communication between atoms, introduced by the global node, does not significantly affect the system's physical behavior in terms of pairwise distances. Further discussions are available in SM Section 9.

Discussion

We develop an accurate and efficient equivariant graph neural network for predicting interatomic potentials and forces. The efficient nature of E²GNN comes from two factors: direct force predictions and modeling only two-body interactions. E²GNN consistently outperforms the prediction performance of the representative baselines across diverse molecular and crystal datasets. Moreover, E²GNN achieves the precision of ab initio MD while maintaining high computational efficiency in simulations involving gas, liquid, and solid systems. Despite its advancements, we would like to say two limitations. First, the introduction of global entities may lead to inaccuracies in certain scenarios where distant parts of the system are incorrectly influenced by global changes, such as two "isolated" molecules in a big box without interaction between them. Second, the forces are predicted directly, which is not energy-conserving, and a thermostat is needed to regulate the temperature[15,36].

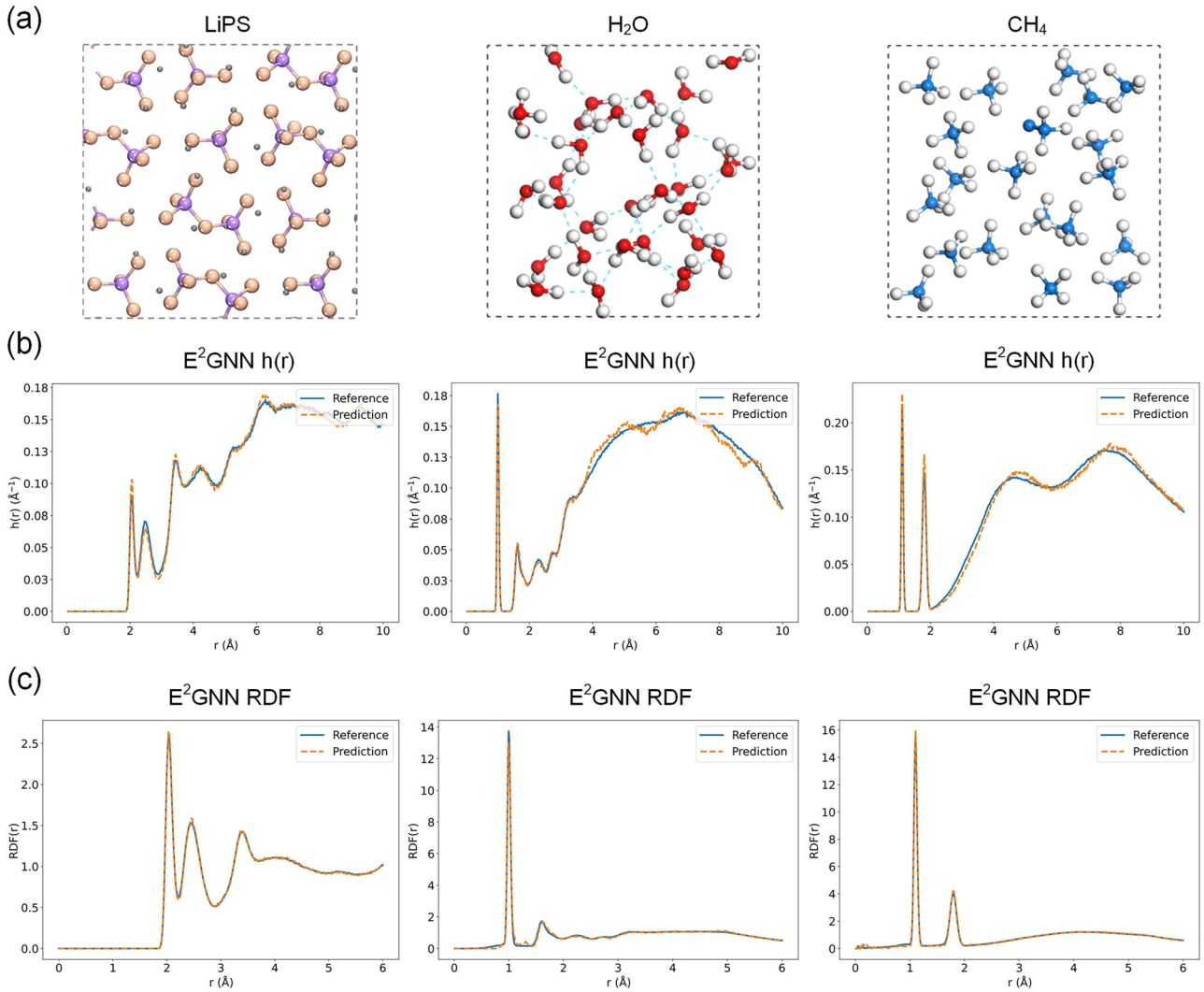

Fig. 6 | MD simulations for LiPS, H_2O, and CH_4. **a** An overview of the three benchmark systems. **b** $h(r)$ for trajectories predicted by the E^2GNN. **c** RDF for trajectories predicted by E^2GNN.

Table 3 | Evaluating E^2GNN and the representative models on MD simulations using LiPS system in terms of forces (meV/Å), stability (ps), MAE of RDF (unitless), and diffusivity (10^{-9} m^2/s)

Model	Force (↓)	Stability (↓)	RDF (↓)	Diffusivity (↓)
DeepPot-SE[a]	40.5	4	0.27	-
SchNet[a]	28.8	50	0.04	0.38
DimeNet[a]	3.2	48	0.05	0.30
ForceNet[a]	12.8	26	0.51	-
GemNet-T[a]	1.3	50	0.04	0.24
GemNet-dT[a]	1.4	50	0.04	0.28
NequIP[a]	3.7	50	0.04	0.34
E^2GNN	1.5	50	0.05	0.21

[a]These results are taken from ref. 36. E^2GNN is evaluated under the same conditions as the reported results.

Data availability

The data that support the findings of this study are available at https://github.com/Shen-Group/E2GNN.

Code availability

The code for E^2GNN, along with detailed instructions, is available at https://github.com/Shen-Group/E2GNN.

Received: 2 January 2024; Accepted: 19 January 2025;
Published online: 26 February 2025

References

1. Batatia, I., Kovacs, D. P., Simm, G., Ortner, C. & Csányi, G. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields. *Adv. Neural Inf. Process. Syst.* **35**, 11423–11436 (2022).
2. Batzner, S. et al. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nat. Commun.* **13**, 2453 (2022).
3. Chen, C. & Ong, S. P. A universal graph deep learning interatomic potential for the periodic table. *Nat. Comput. Sci.* **2**, 718–728 (2022).
4. Xie, F., Lu, T., Meng, S. & Liu, M. Gptff: A high-accuracy out-of-the-box universal ai force field for arbitrary inorganic materials. *Sci. Bulletin* **69**, 3525–3532 (2024).
5. Zhong, Y., Yu, H., Su, M., Gong, X. & Xiang, H. Transferable equivariant graph neural networks for the Hamiltonians of molecules and solids. *npj Comput. Mater.* **9**, 182 (2023).
6. Zhong, Y. et al. Universal machine learning Kohn-sham Hamiltonian for materials. *Chin. Phys. Lett.* **41**, 077103 (2024).
7. Li, H. et al. Deep-learning density functional theory Hamiltonian for efficient ab initio electronic-structure calculation. *Nat. Comput. Sci.* **2**, 367–377 (2022).
8. Wang, Y. et al. Universal materials model of deep-learning density functional theory Hamiltonian. *Sci. Bull.* **69**, 2514–2521 (2024).

9. 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).
10. Yoon, J. & Ulissi, Z. W. Differentiable optimization for the prediction of ground state structures (dogss). *Phys. Rev. Lett.* **125**, 173001 (2020).
11. Deng, B. et al. Chgnet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nat. Machine Intell.* 1–11 (2023).
12. Pablo-García, S. et al. Fast evaluation of the adsorption energy of organic molecules on metals via graph neural networks. *Nat. Comput. Sci.* 1–10 (2023).
13. Li, Y. et al. Local environment interaction-based machine learning framework for predicting molecular adsorption energy. *J. Mater. Inform.* **4**, 4 (2024).
14. Dai, M., Demirel, M. F., Liang, Y. & Hu, J.-M. Graph neural networks for an accurate and interpretable prediction of the properties of polycrystalline materials. *npj Comput. Mater.* **7**, 103 (2021).
15. Hu, W. et al. Forcenet: a graph neural network for large-scale quantum calculations. arXiv preprint arXiv:2103.01436.
16. Xie, T. & Grossman, J. C. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Phys. Rev. Lett.* **120**, 145301 (2018).
17. Schütt, K. et al. Schnet: a continuous-filter convolutional neural network for modeling quantum interactions. *Adv. Neural Inf. Process. Syst.* **30**, 1–11 (2017).
18. Chen, C., Ye, W., Zuo, Y., Zheng, C. & Ong, S. P. Graph networks as a universal machine learning framework for molecules and crystals. *Chem. Mater.* **31**, 3564–3572 (2019).
19. Choudhary, K. & DeCost, B. Atomistic line graph neural network for improved materials property predictions. *npj Comput. Mater.* **7**, 185–192 (2021).
20. Han, J. et al. A survey of geometric graph neural networks: Data structures, models and applications. arXiv preprint arXiv:2403.00485.
21. Gong, X. et al. General framework for e(3)-equivariant neural network representation of density functional theory Hamiltonian. *Nat. Commun.* **14**, 2848 (2023).
22. 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).
23. Brandstetter, J., Hesselink, R., van der Pol, E., Bekkers, E. J. & Welling, M. Geometric and physical quantities improve e(3) equivariant message passing. In: *Proc. International Conference on Learning Representations* (2021).
24. Satorras, V. G., Hoogeboom, E. & Welling, M. E(n) equivariant graph neural networks. In: *Proc. International Conference on Machine Learning*, pp. 9323–9332 (2021).
25. Zitnick, C. L. et al. Spherical channels for modeling atomic interactions. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2022).
26. Passaro, S. & Zitnick, C. L. Reducing so (3) convolutions to so (2) for efficient equivariant gnns. In: International Conference on Machine Learning, pp. 27420–27438 (2023).
27. Yang, Z. et al. Scalable crystal structure relaxation using an iteration-free deep generative model with uncertainty quantification. *Nat. Commun.* **15**, 8148 (2024).
28. Gasteiger, J., Groß, J. & Günnemann, S. Directional message passing for molecular graphs. In: *Proc. International Conference on Learning Representations (ICLR)* (2020).
29. Gasteiger, J., Becker, F. & Günnemann, S. Gemnet: Universal directional graph neural networks for molecules. *Adv. Neural Inf. Process. Syst.* **34**, 6790–6802 (2021).
30. Liu, Y. et al. Spherical message passing for 3d molecular graphs. In: *International Conference on Learning Representations (ICLR)* (2022).
31. Atz, K., Grisoni, F. & Schneider, G. Geometric deep learning on molecular representations. *Nat. Mach. Intell.* **3**, 1023–1032 (2021).
32. Shuaibi, M. et al. Rotation invariant graph neural networks using spin convolutions. arXiv preprint arXiv:2106.09575.
33. Batzner, S., Musaelian, A. & Kozinsky, B. Advancing molecular simulation with equivariant interatomic potentials. *Nat. Rev. Phys.*, 1–2 (2023).
34. Musaelian, A. et al. Learning local equivariant representations for large-scale atomistic dynamics. *Nat. Commun.* **14**, 579 (2023).
35. Batatia, I., Schaaf, L. L., Csanyi, G., Ortner, C. & Faber, F. A. Equivariant matrix function neural networks. In: The Twelfth International Conference on Learning Representations (2024).
36. Fu, X. et al. Forces are not enough: benchmark and critical evaluation for machine learning force fields with molecular simulations. *Trans. Mach. Learn. Res.* (2023).
37. Schütt, K., Unke, O. & Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In: *Proc. International Conference on Machine Learning*, pp. 9377–9388 (2021).
38. Haghighatlari, M. et al. Newtonnet: Newtonian message passing network for deep learning of interatomic potentials and forces. *Digit. Discov.* **1**, 333–343 (2022).
39. 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).
40. Chanussot, L. et al. Open catalyst 2020 (oc20) dataset and community challenges. *ACS Catal.* **11**, 6059–6072 (2021).
41. Tran, R. et al. The open catalyst 2022 (oc22) dataset and challenges for oxide electrocatalysts. *ACS Catal.* **13**, 3066–3084 (2023).
42. Gasteiger, J., Giri, S., Margraf, J. T. & Günnemann, S. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. In: *Proc. Machine Learning for Molecules Workshop, NeurIPS* (2020).
43. Kolluru, A. et al. Open challenges in developing generalizable large-scale machine-learning models for catalyst discovery. *ACS Catal.* **12**, 8572–8581 (2022).
44. Zhang, L. et al. End-to-end symmetry preserving inter-atomic potential energy model for finite and extended systems. *Adv. Neural Inf. Process. Syst.* **31** (2018).

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 62176272), Research and Development Program of Guangzhou Science and Technology Bureau (No. 2023B01J1016), Key-Area Research and Development Program of Guangdong Province (No. 2020B1111100001), Singapore MOE Tier 1 (No. A-8001194-00-00), and Singapore MOE Tier 2 (No. A-8001872-00-00).

Author contributions

L.S. and Z.Y. conceptualized and designed the research. Z.Y. and X.W. conducted the experiments. Z.Y., X.W., Y.L., Q.L., C.Y.-C.C. and L.S. analyzed the data and interpreted the results. Z.Y. and L.S. jointly 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/s41524-025-01535-3.

Correspondence and requests for materials should be addressed to Calvin Yu-Chian Chen or Lei Shen.