Hot-Ham: an Accurate and Efficient $E(3)$-Equivariant Machine-Learning Electronic Structures Calculation Framework

Zhixin Liang(梁智新)†, Yunlong Wang(王云龙)†, Chi Ding(丁驰), Junjie Wang(王俊杰)*, Hui-Tian Wang(王慧田), Dingyu Xing(邢定钰), and Jian Sun(孙建)*

National Laboratory of Solid State Microstructures, School of Physics and Collaborative Innovation Center of Advanced Microstructures, Nanjing University, Nanjing 210093, China

(Received 5 October 2025; accepted manuscript online 17 December 2025)

The combinations of machine learning with *ab initio* methods have attracted much attention for their potential to resolve the accuracy-efficiency dilemma and facilitate calculations for large-scale systems. Recently, equivariant message passing neural networks (MPNNs) that explicitly incorporate symmetry constraints have demonstrated promise for interatomic potential and density functional theory (DFT) Hamiltonian predictions. However, the high-order tensors used to represent node and edge information are coupled through the Clebsch–Gordan tensor product, leading to steep increases in computational complexity and seriously hindering the performance of equivariant MPNNs. Here, we develop high-order tensor machine-learning Hamiltonian (Hot-Ham), an $E(3)$ equivariant MPNN framework that combines two advanced technologies: local coordinate transformation and Gaunt tensor product to efficiently model DFT Hamiltonians. These two innovations significantly reduce the complexity of tensor products from $O(L^6)$ to $O(L^3)$ or $O(L^2 \log^2 L)$ for the max tensor order L, and enhance the performance of MPNNs. Benchmarks on several public datasets demonstrate its state-of-the-art accuracy with relatively few parameters, and applications to multilayer twisted moiré systems, heterostructures, and allotropes showcase its generalization ability and high efficiency. Our Hot-Ham method provides a new perspective for developing efficient equivariant neural networks and would be a promising approach for investigating the electronic properties of large-scale materials systems.

DOI: 10.1088/0256-307X/43/2/020704 CSTR: 32039.14.0256-307X.43.2.020704

1. Introduction. Density functional theory (DFT) has become one of the most popular tools for studying the structure, properties, and reactions of materials at the atomic level. While DFT provides a quantum mechanical framework for electronic structure calculations, solving the Kohn–Sham equations requires significant computational resources, restricting its application to systems with a limited number of atoms. In contrast, empirical tight-binding methods such as the Slater–Koster method [1] can typically be evaluated orders of magnitude faster than first-principles methods due to their simple mathematical form, hence enabling large-scale electronic calculations. However, empirical tight-binding often comes at the cost of reduced accuracy compared to DFT methods and limited transferability across different systems.

Emerging machine learning techniques have been increasingly applied to model electronic structures, [2–15] showing great promise in providing a way to address this accuracy-efficiency dilemma. Among the various approaches, graph neural networks (GNNs) [16–19] have become the dominant choice to characterize the graph structures. Schütt *et al.* [5] proposed a GNN model SchNorb to predict Hamiltonian as a linear combination of spherical harmonics; however, it does not incorporate symmetry as prior knowledge. An important property of Hamiltonian is its equivariant transformation under the Euclidean group in three-dimensional space [$E(3)$ group], which contains translations, rotations, and inversion. Models that explicitly incorporate equivariance constraints are more data-efficient and generalizable, making them promising for complex prediction tasks. PhiSNet, [6] QHNET, [7] and ACE [8] Hamiltonians achieve physical correctness under translation and rotation transformations, but all of them would be struggling with periodic materials due to a lack of parity symmetry. [12] The N-center representation equivariant framework proposed by Nigam [9] achieved $E(3)$-equivariant Hamiltonian using Gaussian process regression (GPR). However, GPR usually has limited generalization ability and is more computationally expensive compared with GNNs such as message passing neural networks (MPNNs). [20] Combining $E(3)$-equivariance constraints with MPNNs, Gong *et al.* [11] and Zhong *et al.* [12] proposed DeepH-E3 and HamGNN respectively and showcased excellent accuracy in Hamiltonian predictions across diverse materials.

Despite the remarkable success of $E(3)$-equivariant neural networks [$E(3)$-ENNs], these models face significant limitations due to the high computational complexity

†These authors contributed equally to this work.

*Corresponding authors. Email: jiansun@nju.edu.cn; wangjunjie@nju.edu.cn

© 2026 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

of equivariant operations that employ Clebsch–Gordan tensor product (CGTP) of irreducible representations (irreps). The CGTP couples tensors with orders up to L to produce new features, leading to a computational complexity of $O(L^6)$ for the full operation. This steeply increasing complexity hinders the application of $E(3)$-ENNs for predicting physical quantities that require high-order tensors. To address this issue, Passaro and Zitnick[21] introduced local coordinate transformation, simplifying CGTP to $SO(2)$ convolution thus reducing the complexity to $O(L^3)$. Recent $E(3)$-equivariant Hamiltonian frameworks DeepH-2[13] and DeePTB[14] have adopted this approach and show state-of-the-art accuracy in Hamiltonian predictions. However, $SO(2)$ convolution is only used for equivariant convolutions, a special case of CGTPs. As there are no specific local coordinate transformations for arbitrary equivariant features, this method is not suitable for equivariant feature interactions (e.g., HamGNN) and equivariant many-body interactions (e.g., MACE[22]). Xin *et al.*[23] proposed another fast and accurate approach for spherical harmonics products, and Luo *et al.*[24] also proposed a similar tensor product named Gaunt tensor products (GTPs), which are more efficient and general as compared with $SO(2)$ convolution. Nonetheless, since GTPs inherently exclude antisymmetric parity, they are hardly utilized in $E(3)$-ENNs, especially for Hamiltonian predictions where antisymmetric tensors are indispensable.

In this work, we propose high-order tensor machine-learning Hamiltonian (Hot-Ham), an $E(3)$-equivariant MPNN framework designed for efficient Hamiltonian representation. By combining local coordinate transformations and GTPs, our framework not only effectively leverages their strengths but also compensates for their limitations, reducing tensor products from $O(L^6)$ to $O(L^3)$ or $O(L^2 \log^2 L)$ and establishing an efficient $E(3)$-ENN for electronic structure calculation. We benchmark Hot-Ham on several public datasets and compare its performance with other Hamiltonian models. Our results show that Hot-Ham achieves superior performance while maintaining a significantly smaller parameter size (within $2M$) compared to most models. To demonstrate the generalization ability and efficiency, we train Hot-Ham on non-twisted MoS_2, graphene/hexagonal boron nitride (h-BN) heterostructures and phosphorus allotropes, and test it on various structures outside of the datasets. The accuracy shows that our model is able to accurately predict electronic structure even though the crystal structure is not included in the training set, with computational costs much lower than DFT. The accuracy, generalization, and efficiency of Hot-Ham significantly mitigate the accuracy-efficiency trade-off dilemma of DFT, opening up new possibilities for large-scale electronic structure calculations and novel functional material discovery.

2. Results.

2.1. Equivariant Hamiltonian. The properties of physical systems generally exhibit symmetries under transformations of a set of atoms, leading to the concept of equivariance naturally. Formally, a function $f : X \to Y$ is equivariant for X and Y with respect to group G if $f(g_X x) = g_Y f(x)$ for $\forall g \in G$, where g_X and g_Y are group representations on X and Y. ENNs are guaranteed to preserve equivariance under a change of coordinates because they are composed of equivariant functions.

In GNNs, the structure is represented as a graph, where each node corresponds to an atom, and each edge represents a connection between atoms within a cutoff radius r_c. Given a set of atoms, the two key problems of designing ENNs are how to encode the atom positions $\{\boldsymbol{r}_i\}$ and chemical elements $\{\boldsymbol{Z}_i\}$ as equivariant features for nodes and edges, and how to relate these features to equivariant Hamiltonians.

Traditionally, DFT Hamiltonians can be calculated on various bases, such as plane-wave basis,[25–27] linear combination of atomic orbitals (LCAO) basis,[28–31] and so on. Specifically, the LCAO basis has the form $\phi_{nlm}(\boldsymbol{r} - \boldsymbol{r}_i) = R_{nl}(|\boldsymbol{r} - \boldsymbol{r}_i|)Y_{lm}(\boldsymbol{r} - \boldsymbol{r}_i)$, where i is the atom index, R_{nl} and Y_{lm} are radial function and spherical harmonics respectively with projectors' multiplicity n, angular momentum quantum number l, and magnetic quantum number m. On the one hand, this form restricts the interaction calculations within a cutoff distance, which is consistent with the finite receptive field of ENNs. On the other hand, spherical harmonics, which are also referred to as spherical tensors, satisfy rotation and inversion equivariance [$O(3)$ group], and have been widely utilized to represent equivariant features inside ENNs.[22,32–34] Therefore, among various kinds of bases, LCAO-basis Hamiltonians are suitable learning targets for machine learning Hamiltonian model. According to Wigner–Eckart theorem, the LCAO-basis Hamiltonian element $H_{ij,n_1l_1m_1n_2l_2m_2} = \langle i, n_1l_1m_1|\hat{H}|j, n_2l_2m_2\rangle$ between i, j atoms can be decomposed into several learnable spherical tensors $x_{ij,c_3l_3m_3p_3}$:

$$\begin{aligned}&\langle i, n_1l_1m_1|\hat{H}|j, n_2l_2m_2\rangle \\ &= \sum_{l_3=|l_1-l_2|}^{l_1+l_2} \sum_{m_3=-l_3}^{l_3} C_{l_1m_1l_2m_2}^{l_3m_3} x_{ij,c_3l_3m_3p_3}, \end{aligned} \tag{1}$$

where c_3 is the channel index that is determined by the n and l of two atomic orbitals, and $p_3 \in [1, -1]$ is the parity index. $C_{l_1m_1l_2m_2}^{l_3m_3}$ is the Clebsch–Gordan coefficient. The atomic orbitals basis for ith atom $|i, n_1l_1m_1\rangle = \phi_{n_1l_1m_1}(\boldsymbol{r} - \boldsymbol{r}_i)$ has a parity of $p_1 = (-1)^{l_1}$. So the parities of $H_{ij,n_1l_1m_1n_2l_2m_2}$, and thus $x_{ij,c_3l_3m_3p_3}$, are determined by the two atomic orbitals basis as $p_3 = (-1)^{l_1+l_2}$. $x_{ij,c_3l_3m_3p_3}$ represents the equivariant feature of atom for onsite block if $i = j$, or edge for hopping block if $i \neq j$. Hence, the aim of ENNs is to predict high-order tensor features that comprise a direct sum of irreps of the $O(3)$ for all nodes and edges.

Chemical elements and interatomic distances are invariant physical quantities and are used to construct 0-order tensors, i.e., scalars. The relative positions of atoms are mapped into spherical harmonics to serve as high-order tensors, which automatically ensure the invariance of ENNs under translation. After a series of equivariant operations, the final equivariant features are used to reconstruct Hamiltonians in a block-wise manner through

Eq. (1). It should be noted that the equivariant operations have to be carefully designed, not only to preserve $E(3)$ equivariance but also to achieve high efficiency for the neural networks. One of the main equivariant operations is the $E(3)$ linear transformations. The linear operations mix the features between channels in the same irreps. For the common feature representation $X^L = \bigoplus_{l=0}^{L} x_l$, which represents the direct sum of spherical tensors x_l up to order L, the linear transformations can be conducted with a complexity of $O(L^2)$. The other main equivariant operations are tensor products. These operations couple tensors x_1 and x_2 in different irreps through the Clebsch–Gordan coefficient and yield new tensors x_3 that are $E(3)$-equivariant:

$$\begin{aligned} x_{3,l_3m_3} &= (x_1 \otimes_{\mathrm{CG}} x_2)_{l_3m_3} \\ &= \sum_{m_1m_2} C^{l_3m_3}_{l_1m_1l_2m_2} x_{1,l_1m_1} x_{2,l_2m_2}. \end{aligned} \tag{2}$$

Such a tensor product path's complexity scales with $O(L^3)$. However, when applying Eq. (2) to features like X^L, since the number of paths is as high as $O(L^3)$, the complete tensor products will lead to an $O(L^6)$ complexity, which seriously hinders the application of ENNs, especially for Hamiltonian predictions where high-order tensors are essential to represent atomic orbitals and their interactions. Therefore, new approaches to achieve efficient tensor products are demanding.

2.2. Efficient Tensor Product Operations. The $SO(2)$ convolution is one of the approaches to accelerating the tensor product operations. In the field of ENNs, filters can be defined as spherical harmonics acting on relative positions $\boldsymbol{r}_{ij}$. For the convolution operation, i.e., tensor product between feature x_{1,l_1} and filter $x_{2,l_2} = Y_{l_2}(\boldsymbol{r}_{ij})$, the complexity can be reduced through local coordinate transformation. By rotating the embeddings' primary axis to align with the edge vectors, the filter $Y_{l_2m_2}(\boldsymbol{r}_{ij})$ becomes sparse: $Y_{l_2m_2}(\boldsymbol{r}_{ij}) = 0$ if $m_2 \neq 0$. This eliminates the summation over m_2, reducing Eq. (2) to a 2D matrix multiplication. By rearranging the Clebsch–Gordan coefficients and relevant weights, the tensor products between rotated features like X^L and filters with orders up to L are simplified to $O(L)$ 2D matrix multiplications, which accounts for an $O(L^3)$ cost. Additionally, an $O(L^2)$ operation is needed for each irrep among X^L to rotate to align with the edge vector or rotate back, accounting for two extra $O(L^3)$ costs.

GTP is another more efficient and general approach. Gaunt coefficients[35] $G^{l_3m_3}_{l_1m_1l_2m_2}$ are defined as the integrals of three spherical harmonics products and are related to Clebsch–Gordan coefficients via constant factors $\tilde{C}^{l_3}_{l_1l_2}$ that are independent of the magnetic quantum number: $G^{l_3m_3}_{l_1m_1l_2m_2} = \tilde{C}^{l_3}_{l_1l_2} C^{l_3m_3}_{l_1m_1l_2m_2}$. Luo *et al.*[24] propose new perspectives that GTPs calculate the coefficients of spherical functions based on spherical harmonics,

$$\begin{aligned} &\sum_{l_1}^{L_1}\sum_{l_2}^{L_2} (x_{1,l_1} \otimes_{\mathrm{Gaunt}} x_{2,l_2})_{l_3m_3} \\ &= \int_0^{2\pi}\int_0^{\pi} F_{L_1}(\theta,\psi) F_{L_2}(\theta,\psi) Y_{l_3m_3}(\theta,\psi) \sin\theta \mathrm{d}\theta\mathrm{d}\psi, \end{aligned} \tag{3}$$

for which $F_{L_i}(\theta,\psi) = \sum_{l=0}^{L_i}\sum_{m=-l}^{l} x_{i,lm} Y_{lm}(\theta,\psi)$ is a square-integrable spherical function, and appropriate base transformations will be helpful to parallel these operations. An illustration is presented in Fig. 1(a). By changing the spherical harmonics bases into 2D Fourier bases, the spherical function $F(\theta,\psi) = F_{L_1}(\theta,\psi)F_{L_2}(\theta,\psi)$ will be simplified into 2D convolution, which can be accelerated via 2D fast Fourier transform (FFT) in $O(L^2 \log L)$. Finally, the coefficients of $F(\theta,\psi)$ are converted back to spherical harmonic representations. These base transformations account for an $O(L^3)$ cost. We call it GTP(2D-FB) in this paper to distinguish it from the method GTP(sphere-grid) proposed by Xie *et al.*[36] through spherical convolution, as shown in Fig. 1(b). Features are first projected onto a sphere grid through an inverse FFT. Following an $O(L^2)$ element-wise product, a FFT is performed to convert back to spherical harmonics. The two FFTs can theoretically achieve a complexity of $O(L^2 \log^2 L)$ at best.[37] Similarly, the coefficients of both GTP methods can be further sparse via local coordinate transformation in the

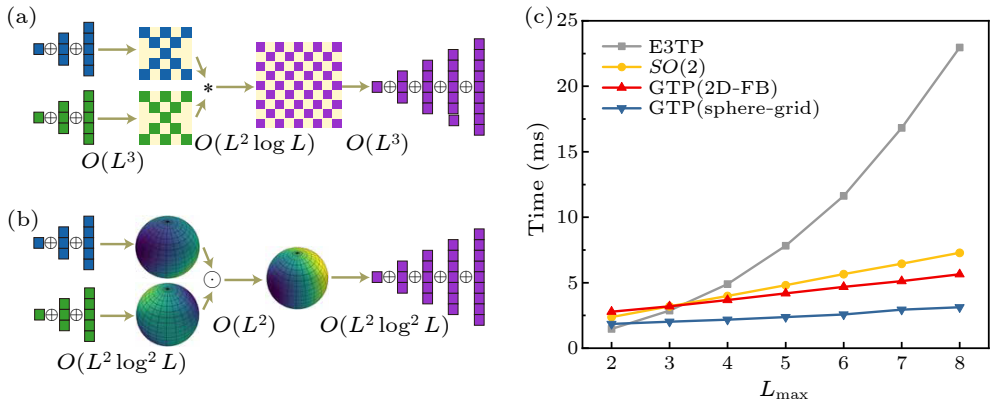

Fig. 1. Illustration of the GTP method and comparison of tensor product efficiency. (a) GTP(2D-FB) method. Spherical harmonic features are transformed into 2D Fourier basis representation. After FFT convolution, they are transformed back into spherical harmonic representation. (b) GTP(sphere-grid) method. Spherical harmonic features are projected onto a sphere grid via inverse FFT, followed by an element-wise product, and finally converted back to spherical harmonic representation. (c) Comparison of efficiency for CGTP (implemented by e3nn, labeled as E3TP), $SO(2)$ convolution, GTP(2D-FB), and GTP(sphere-grid).

case of convolution. However, it should be noted that there is a difference between GTP and CGTP, because $\tilde{C}^{l_3}_{l_1 l_2}$ vanish when $l_1 + l_2 + l_3 = 2k + 1$, where k is a non-negative integer; thus leading to the lack of antisymmetric tensors in GTP. Nonetheless, GTPs still adhere to $E(3)$ equivariance requirements.

We conduct experiments on the convolution operations using these four tensor products to demonstrate their efficiency, as shown in Fig. 1(c). The original CGTP implemented by the e3nn package is unexpectedly the most time-consuming, while GTPs, especially GTP(sphere-grid), achieve the highest efficiency. These results, as well as the fact that GTP is suitable for various tensor products not only for convolution, indicate that GTP is a promising operation to achieve high-efficiency equivariant MPNNs.

2.3. Equivariant Message Passing Neural Network. The Hot-Ham model architecture is illustrated in Fig. 2. Atomic number Z_i, interatomic distance $|\boldsymbol{r}_{ij}|$ and direction information $Y(\boldsymbol{r}_{ij})$ are embedded to generate initial node features $v_i^{(0)}$ and edge features $e_{ij}^{(0)}$. Features are structured as a direct sum of symmetric tensors $x_{c_0 lm,(-1)^l}$ and antisymmetric tensors $x_{c_1 lm,(-1)^{l+1}}$. Since only symmetric tensors can be generated by GTP, we need to introduce antisymmetric tensors at least once through CGTP. Given that including antisymmetric tensors in intermediate layers would result in about double computational costs, we choose to perform CGTP in the last layer through $SO(2)$ convolution. Node and edge symmetric tensor features are encoded and aggregated to update iteratively in each GTP convolution layer, then are extended to include antisymmetric tensor features in the last $SO(2)$ convolution layer. Finally, these features are transformed into Hamiltonian matrix block H_{ij} through Wigner-Eckart theorem in the Readout layer.

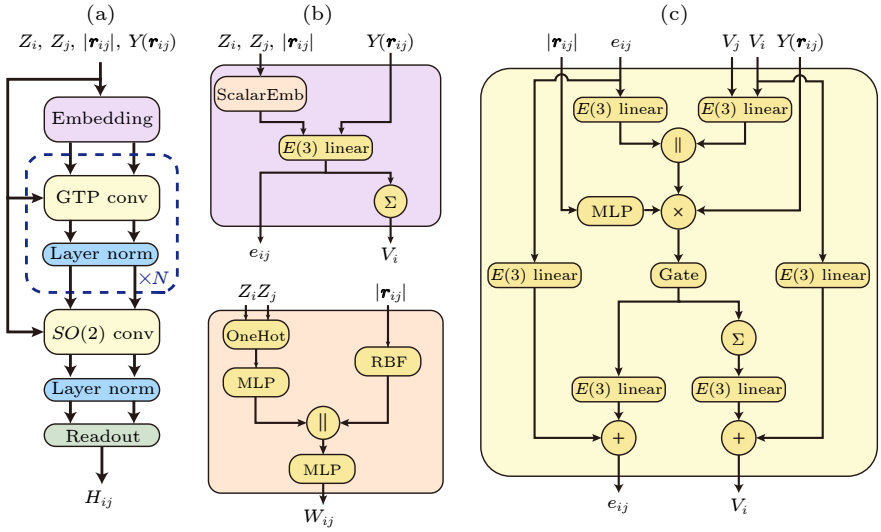

Fig. 2. The architecture of Hot-Ham. (a) The overall architecture of Hot-Ham. Atom species and relative positions are embedded into initial node and edge features. After several iterative updates in convolution layers, the final node and edge features are used to construct Hamiltonians. (b) The embedding layer. The scalar information including atom species Z_i, Z_j and interatomic distance $|\boldsymbol{r}_{ij}|$ is embedded through the ScalarEmd block into weight w_{ij}, and then is used to weight the spherical harmonic functions $Y(\boldsymbol{r}_{ij})$. (c) The convolution layer. Node and edge features are concatenated and then multiplied with $Y(\boldsymbol{r}_{ij})$ to serve as messages. Since both GTP and $SO(2)$ can be used for this operation, we did not explicitly distinguish between them, except for labeling them in panel (a). After the gate (activation function), messages are used to generate new node and edge features simultaneously.

2.3.1. Feature Initialization. As shown in Fig. 2(b), the edge features are defined as weighted spherical harmonics through a linear transformation, and node features are the mean of edge features,

$$\begin{aligned} e^{(0)}_{ij,lmc} &= w_{ij,lc}(|\boldsymbol{r}_{ij}|)Y_{lm}(\boldsymbol{r}_{ij}), \\ v^{(0)}_{i} &= \frac{1}{|N(i)|}\sum_{j\in N(i)} e^{(0)}_{ij}, \end{aligned} \tag{4}$$

where $N(i)$ are neighbor atoms of atom i. $w_{ij,lc}$ are weights that contain information about the species of two connected atoms and their interatomic distance. Specifically, Z_i and Z_j are mapped by one-hot encoding and then passed through a learnable multi-layer perceptron (MLP) into two species vectors, and $|\boldsymbol{r}_{ij}|$ is expanded by radial basis functions (RBFs), such as Gaussian, Bessel, and Chebyshev bases. Subsequently, these species vectors and distance expansions are concatenated and transformed by MLP into $w_{ij,lc}$.

2.3.2. Convolution. In convolution layers, local environment information including connected node features $v_i^{(t)}$ and edge feature $e_{ij}^{(t)}$ are used to encode edge messages through the tensor products between the weighted concatenations $C^{(t)}_{ij,cl_1m_1}$ of features and weighted filters $F^{(t)}_{ij,l_2m_2}$,

$$\begin{aligned} C^{(t)}_{ij,c_{\rm out}l_1m_1} &= w^{1,(t)}_{ij,c_{\rm in}c_{\rm out}l_1}(|\boldsymbol{r}_{ij}|)\left(v_i^{(t)}\middle\|v_j^{(t)}\middle\|e_{ij}^{(t)}\right)_{c_{\rm in}l_1m_1}, \\ F^{(t)}_{ij,l_2m_2} &= w^{2,(t)}_{ij,l_2}(|\boldsymbol{r}_{ij}|)Y_{l_2m_2}(\boldsymbol{r}_{ij}), \\ m^{(t)}_{ij,c_{\rm out}l_3m_3} &= \sum_{l_1m_1l_2m_2} C^{l_3m_3}_{l_1m_1l_2m_2} C^{(t)}_{ij,c_{\rm out}l_1m_1} F^{(t)}_{ij,l_2m_2}, \end{aligned} \tag{5}$$

where $w^{1,(t)}_{ij,c_{\rm in}c_{\rm out}l}$ and $w^{2,(t)}_{ij,l_2}$ are distance-dependent weight

functions, and $\|$ denotes concatenation. To introduce nonlinearity without breaking $E(3)$ equivariance, a gate (activation function) must be carefully designed. In Hot-Ham, we use a gate like Ref. [38], where scalar and pseudoscalar components are nonlinear by SiLU and tanh respectively, while others are scaled by learnable scalars that are nonlinear by sigmoid. Messages are then used to update edge and node features simultaneously. New edge features are updated with the residual net-like architecture,[33]

$$\tilde{e}_{ij,clm}^{(t+1)} = w_{0,lc}^{(t)} e_{ij,clm}^{(t)} + w_{1,lc}^{(t)} \mathrm{Gate}\left(m_{ij,clm}^{(t)}\right). \tag{6}$$

For node features, messages are first propagated to each node along edges and then aggregated by the mean function. Similarly, the old node features are combined with the messages to form the new node features,

$$\tilde{v}_{i,clm}^{(t+1)} = w_{2,cl}^{(t)} v_{i,clm}^{(t)} + w_{3,clm}^{(t)} \cdot \left[\frac{1}{|N(i)|} \sum_{j \in N(i)} \mathrm{Gate}\left(m_{ij,clm}^{(t)}\right)\right]. \tag{7}$$

2.3.3. Layer Normalization. Normalization is a crucial technique in deep learning models used to stabilize and enhance training. Layer normalization is one of the most common methods applied in ENNs due to its independence of batch size, which is typically small in Hamiltonian model training. To preserve equivariance, feature components belonging to different irreps are normalized separately. For scalar features $\tilde{f}_{z,0}^{(t)}$, we use layer normalization as described in Refs. [11,32],

$$f_{z,c00}^{(t)} = \gamma_{c0} \frac{\tilde{f}_{z,c0}^{(t)} - \mu_{00}}{\sigma_0 + \varepsilon} + \beta_{c0}, \tag{8}$$

where z represents atom index i for $\tilde{v}_{i,clm}^{(t)}$ or edge index ij for $\tilde{e}_{ij,clm}^{(t)}$, $\mu_{lm} = 1/(N_Z N_c) \sum_{Z=1,c=1}^{N_Z,N_c} \tilde{f}_{Z,clm}^{(t)}$ and $\sigma_l = \sqrt{1/(N_Z N_c) \sum_{Z=1,c=1}^{N_Z,N_c} |\tilde{f}_{Z,clm}^{(t)} - \mu_{lm}|^2}$ are mean and standard deviation of features with degree l. ε is a small number used to avoid numerical instability when σ_0 is close to zero. γ_{cl} and β_{cl} are learnable parameters for affine transformation. While for pseudoscalar or tensor with $l > 0$, we simply subtract the mean as

$$f_{z,clm}^{(t)} = \tilde{f}_{z,clm}^{(t)} - \mu_{lm}. \tag{9}$$

2.3.4. Readout. The Hamiltonian matrix block $H_{ij,l_1 l_2}$ between atom i with orbital angular momentum l_1 and atom j with orbital angular momentum l_2 takes the form of a direct product: $l_1 \otimes l_2$, a $(2l_1+1)\times(2l_2+1)$ matrix with parity $p = (-1)^{l_1+l_2}$. After several layers of convolution iterations, the final features have to be passed through a linear layer to rearrange into a form $\bigoplus_{l=|l_1-l_2|}^{l_1+l_2} x_{ij,lp}$, and then transform to $H_{ij,l_1 l_2}$ through Eq. (1). Incorporating the prior Hermitian property for Hamiltonian will be beneficial in reducing loss. To enforce the Hamiltonian to be Hermitian, we define the final Hamiltonian matrix block as $\tilde{H}_{ij} = (1/2)(H_{ij} + H_{ji}^*)$.

2.4. Performance of Hot-Ham. To benchmark the accuracy of our models, we apply Hot-Ham to public datasets,[11] including monolayer graphene, monolayer MoS_2, and bilayer graphene, as reported in Ref. [10]. For each dataset, we train two models using GTP(2D-FB) and GTP(sphere-grid) respectively in the GTP convolution layers and compare the results with other models. To further demonstrate the generalization and efficiency of our approach, we also apply Hot-Ham to study the electronic structure of multilayer twisted MoS_2, graphene/h-BN heterostructures, and phosphorus allotropes.

2.4.1. Benchmark Accuracy. We compare the mean absolute error (MAE) of Hamiltonian matrix elements and the number of training parameters for Hot-Ham with other models, including DeepH-E3, HamGNN, DeepH-2, and DeePTB. The results for the monolayer graphene and monolayer MoS_2 datasets, which are commonly used to evaluate the accuracy of $E(3)$-equivariant Hamiltonian frameworks, are shown in Table 1. The results for the bilayer graphene dataset are provided in the Supplementary Materials (SM). Both GTP implementations share the same tensor product paths, so they are expected to achieve comparable accuracy. This is confirmed by the results of Hot-Ham. Our models demonstrate state-of-the-art accuracy across all systems, achieving 0.08, 0.12, and 0.15 meV MAEs for monolayer graphene, monolayer MoS_2, and bilayer graphene respectively. Notably, compared to the models achieving the best performance in each system, our models require significantly fewer parameters.

Table 1. Comparison of the results by DeepH-E3, HamGNN, DeepH-2, DeePTB, and Hot-Ham on Hamiltonians of monolayer graphene and monolayer MoS_2. (MAEs are in units of meV. Parentheses indicate the number of parameters. For each model, we only show their best performance found in previous work.)

	DeepH-E3[14]	HamGNN	DeepH-2[13]	DeePTB[14]	Hot-Ham(2D-FB)	Hot-Ham(sphere-grid)
Monolayer graphene	0.28(4.5M)	0.17(4.3M)	0.12	0.14(4.5M)	0.08(0.9M)	0.07(0.9M)
Monolayer MoS_2	0.46(1.0M)	0.37(4.3M)	0.21	0.14(4.5M)	0.12(1.9M)	0.12(1.9M)

2.4.2. Multilayer Twisted MoS_2. 2D multilayer materials have garnered significant attention due to their unique electronic, optical, and mechanical properties, which arise from their tunable interlayer coupling under different twist angles and complex stacking configurations. The twist angles between layers can lead to a range of fascinating physical phenomena, such as unconventional superconductivity and nonlinear optical effects.[39] Additionally, the stacking configuration of each adjacent layer can be independently modulated, as the weak interlayer coupling, dominated by van der Waals interactions, allows for considerable flexibility.[40] However, fabricating multilayer materials with precise control over twist angles is not an easy task for experiments, and performing large-scale electronic structure calculations is impracticable for DFT. In the following, we demonstrate that Hot-Ham offers a general and efficient approach to studying electronic structures by investigating multilayer twisted MoS_2.

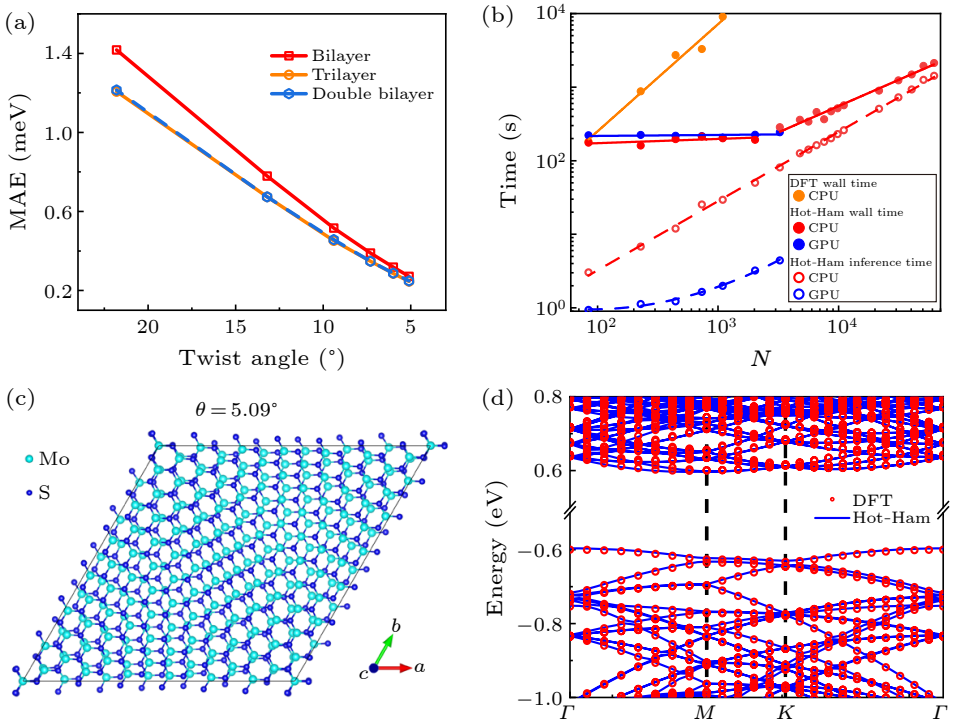

Fig. 3. Application of Hot-Ham to multilayer MoS_2. (a) Hamiltonian MAE for bilayer, trilayer, and double bilayer under different twist angles. (b) DFT wall time and our model's wall time and inference time using CPU and GPU. (c) Structure of double bilayer MoS_2 containing 1524 atoms with twist angle $\theta = 5.09°$. (d) Band structures for the structure in panel (c) predicted by DFT (OpenMX) and Hot-Ham.

Through training on bilayer and trilayer non-twisted MoS_2, our models achieve an accuracy of 0.20 meV. To demonstrate the generalization ability, we applied the trained model to multilayer twisted MoS_2 systems, including bilayer, trilayer, and double bilayer structures. For trilayer MoS_2, we rotated the second layer while keeping the third layer aligned with the first. In the case of double bilayer MoS_2, we rotated the upper two layers, as schematized with one of the rotation angles in Fig. 3(c). The MAEs are shown in Fig. 3(a). Despite the training set not containing any four-layer structures, our model still predicts Hamiltonians with an accuracy comparable to those of trilayers. The MAEs reach relatively large values at a 21.79° twist angle, the maximum angle which is typically used for modeling twist systems; however, their values of 1.2–1.4 meV are still sufficient to accurately describe the electronic structure. The MAEs decrease gradually as the twist angle decreases, approaching the values of non-twisted systems. This can be attributed to smaller deviation from the training set as the twist angle decreases. In addition to its powerful generalization ability, our model also exhibits high efficiency. DFT wall time and our model's wall time and inference time are displayed in Fig. 3(b). The CPU and GPU calculations are performed in one node equipped with Intel Xeon Gold 6140 processors and one NVIDIA GeForce RTX 4090, respectively. The inference times of the CPU and GPU are close. The CPU inference further expands the scale to 10^4, in a time cost that increases linearly with a slope less than DFT calculations when $N > 3000$. We can easily derive band structures by diagonalizing the predicted Hamiltonians. Figure 3(d) illustrates the band structure near the Fermi level for double bilayer MoS_2 with a twisted angle of 5.09°. The results of our model exhibit a high degree of agreement with those obtained from DFT calculations.

2.4.3. Heterostructures. By stacking 2D materials with different properties in an incommensurate manner, heterostructures exhibit novel physical effects that are significantly distinct from the intrinsic properties of individual materials. This provides a new degree of freedom to manipulate properties and band structures, thereby enriching the properties of 2D material systems. Although the lattice mismatches introduced by incommensurability and the possible twist angles make the heterostructures rather complex and rich, our model, trained on graphene/h-BN systems for example, will show its capability to accurately describe the electronic structures.

Similarly, our model is trained on the non-twisted graphene/h-BN but with a mismatch of 1.79%[41,42] corresponding to the difference in lattice constants between graphene and h-BN. These structures are generated from three stacking configurations,[43] one of which is AA and another two are Bernal arrangements (AB and AB'). Figure 4(b) demonstrates the good agreement on band structure for a Bernal arrangement configuration [Fig. 4(a)]. To test the generalization ability of Hot-Ham on unseen configurations, we generate 100 heterostructures with various mismatches and twisted angles within 500 atoms using the VASPKIT package[44] for band structure calculations. As shown in Fig. 4(c), the predicted band eigenvalues range from −1.5 eV to 1.5 eV for all the 100 structures, achieving a MAE of 5.46 meV, while the MAE for Hamiltonians

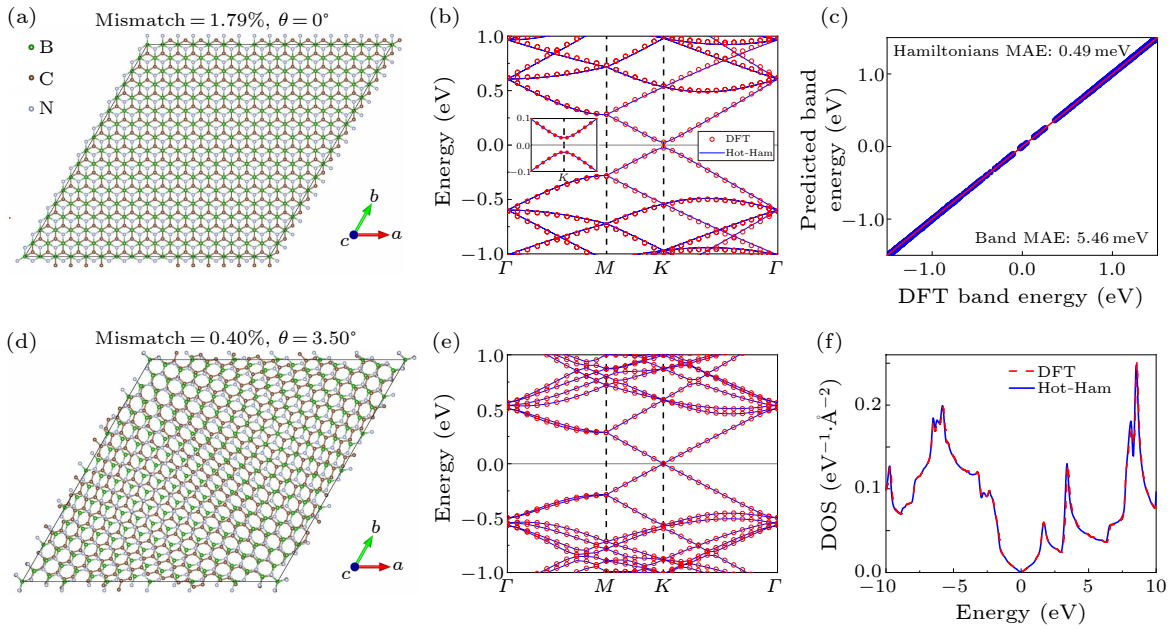

Fig. 4. Performance of Hot-Ham on graphene/h-BN heterostructures. (a) The 16 × 16 supercell of a Bernal arrangement graphene/h-BN heterostructure. (b) The band structure of panel (a). (c) Comparison of band eigenvalues within −1.5 to 1.5 eV for the 100 heterostructures with various mismatch and twisted angles. The MAE of their Hamiltonians is also displayed. (d) Structure of graphene/h-BN containing 1022 atoms (518 C atoms, 252 B/N atoms) with mismatch being 0.4% and twist angle $\theta = 3.50°$. (e) and (f) The band structure and DOS for structure of panel (d) predicted by DFT (OpenMX) and Hot-Ham.

is only 0.49 meV. This accuracy is sufficient to capture significant features, such as the relatively large band gap induced by inversion symmetry breaking, as shown in the insert of Fig. 4(b). Finally, we test our model on a larger system that contains 1022 atoms [Fig. 4(d)]. The predicted band structure [Fig. 4(e)] and density of states [DOS, Fig. 4(f)] agree well with results obtained from OpenMX. These again indicate that Hot-Ham can serve as a robust and powerful approach to exploring the electronic structures of a wide variety of configurations, even though it was trained on a limited dataset.

2.4.4. Phosphorus Allotropes. In addition to 2D materials, we also evaluated Hot-Ham's performance on bulk systems using phosphorus allotropes as an example. We have collected 10 allotropes from the Materials Project,[45] which contains both 2D and bulk systems. Among these structures, the violet phosphorus that has been experimentally synthesized recently is found to be the most stable allotrope at ambient pressure[46] and is thought of as a promising semiconducting material for photonic and electronic applications. Therefore, we use the violet phosphorus to demonstrate Hot-Ham's generalization ability, while the remaining 9 allotropes are used for training.

Our model achieves an accuracy of 0.68 meV MAE for the Hamiltonian matrix on the test set. Then we verify the accuracy on violet phosphorus. The violet phosphorus shown in Fig. 5(a) has 84 atoms in its unit cell, featuring tubular $P2[P8]P2[P9]$ strands arranged perpendicularly. The MAE of the Hamiltonian matrix is only 0.89 meV, enabling our model to accurately reproduce the band structure in Fig. 5(b). This example demonstrates that Hot-Ham can effectively represent features of atoms and bonds in diverse chemical environments, showcasing its potential for applications in high-throughput material discovery, such as crystal structure searches[47] and functional materials predictions.

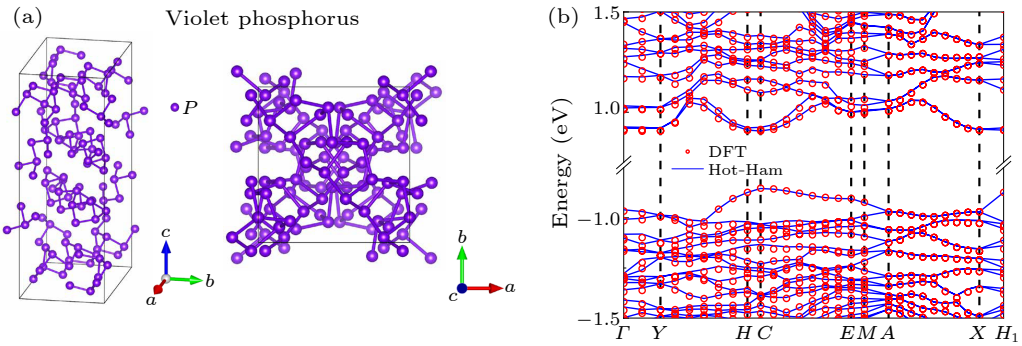

Fig. 5. Predicted electronic structure on phosphorus allotropes. (a) Crystal structure of violet phosphorus. (b) Band structure calculated by DFT (OpenMX) and Hot-Ham for the structure that is not in the training dataset.

3. Discussion. In this work, we develop Hot-Ham, an $E(3)$-equivariant MPNN that combines local coordinate transformation and GTP to efficiently model DFT Hamiltonians from material structures. Through a local coordinate transformation, $SO(2)$ convolution significantly reduces the CGTP computational cost from $O(L^6)$ to $O(L^3)$ for the convolution operations. While GTP provides a more efficient and general approach to implement tensor products, either through convolving in 2D Fourier bases with $O(L^3)$ or by spherical convolution with $O(L^2 \log^2 L)$, it does so at the expense of the absence of antisymmetric tensors. Our method not only retains the advantage of high efficiency of GTP but also compensates for the antisymmetric tensors through $SO(2)$ convolution, achieving an accurate, generalizable, and efficient $E(3)$-equivariant framework for electronic structure calculations for large systems. The comparison with other $E(3)$-ENN models demonstrates Hot-Ham's state-of-the-art accuracy with a significantly smaller parameter size compared to most models. The experiments on MoS_2, heterostructures, and phosphorus allotropes show Hot-Ham's good generalization ability to structures outside of the training sets. Recently, the development of universal Hamiltonian models [48–50] has attracted significant attention. Among the core architectural components, convolutions and many-body interactions have emerged as key bottlenecks limiting model efficiency but can be further optimized via $SO(2)$ convolution and GTPs. While our model currently lacks many-body interactions, incorporating them via GTPs is natural and is expected to further enhance the model's performance and pave the way for the development of universal models. In the future, it should be very valuable to expand the application scope of Hot-Ham to overcome challenges faced by traditional *ab initio* methods. For example, one can derive forces from predicted Hamiltonians, allowing for the explicit incorporation of electronic effects in molecular dynamics simulations. In addition, by utilizing automatic differentiation techniques, one would be able to investigate electron–phonon interactions for large-scale systems. Furthermore, predicting orthogonal-base Hamiltonians to interface with linear scaling quantum transport methodologies [51,52] is also a possible direction. In summary, our Hot-Ham method is a promising framework with great potential for large-scale electronic calculations which are fundamentally important for designing electronic devices.

4. Materials and Method.

4.1. Dataset. The Hamiltonians and overlap matrices are all calculated by OpenMX using the Perdew–Burke–Ernzerhof [53] functional and norm-conserving pseudopotential. The poly-alpha-olefins employed for B, C, N, P, S, Mo, Ti, Nb are B7.0-s2p2d1, C6.0-s2p2d1, N6.0-s2p2d1, P7.0-s2p2d1, S7.0-s2p2d1, Mo7.0-s3p2d2, Ti7.0-s3p2d2f1, and Nb7.0-s3p2d2f1, respectively. We use a 3×3 supercell and 6.50 Å interlayer spacing for multilayer MoS_2; a 4×4 supercell and 3.22 Å interlayer spacing for graphene/h-BN; enlarge the phosphorus allotropes to obtain supercells with 32 to 64 atoms; and use a $2\times2\times2$ supercell for Ti_3Nb. In addition to applying random perturbations at 300 K using the Phonopy package, we also performed random interlayer slides for multilayer MoS_2 and graphene/h-BN up to their lattice constants. The structure numbers are 350 for multilayer MoS_2; 450 for graphene/h-BN; 450 for phosphorus allotropes; and 90 for Ti_3Nb. For each system, the dataset is split into training, testing, and validation sets in a ratio of 3:1:1. For structures used in band structure predictions, twisted multilayer MoS_2 was modeled using the ASE package. [54] The graphene/h-BN heterostructures with no more than 500 atoms were generated through the VASPKIT package by setting a maximum mismatch of 1.79%. While the twisted heterostructure containing 1022 atoms was also modeled using the ASE package.

4.2. Details of Neural Network and Training. In our model, we use the Chebyshev basis to expand the interatomic distance $|\boldsymbol{r}_{ij}|$,

$$\mathrm{RBF}(|\boldsymbol{r}_{ij}|)_k = \frac{1}{2}\Big\{T_k\Big[2\Big(\frac{|\boldsymbol{r}_{ij}|}{r_\mathrm{c}}-1\Big)^2\Big]+1\Big\}f_\mathrm{c}(|\boldsymbol{r}_{ij}|),$$

where r_c is the cutoff distance, T_k is the kth order Chebyshev polynomial of the first kind, and f_c is the cutoff function

$$f_\mathrm{c}(|\boldsymbol{r}_{ij}|) = \frac{1}{2}\Big[1+\cos\Big(\pi\frac{|\boldsymbol{r}_{ij}|}{r_\mathrm{c}}\Big)\Big].$$

The features passed to gate have the form

$$x = \Big(\bigoplus\nolimits_{c_1} x_{1,c_1001}\Big)\bigoplus\Big(\bigoplus\nolimits_{c_2l_2m_2} x_{2,c_2l_2m_2p_2}\Big).$$

Then it is nonlinear as

$$\mathrm{Gate}(x)_{clmp} = \begin{cases} \mathrm{SiLU}(x_{2,clmp}), & l=0,\ p=1,\\ \tanh(x_{2,clmp}), & l=0,\ p=-1,\\ \mathrm{sigmoid}(x_{1,c00})x_{2,clmp}, & l>0. \end{cases}$$

The target of the neural network is to minimize the following loss function:

$$L(H) = \frac{1}{N_\mathrm{e}}\sum_{i=1}^{N}\sum_{\alpha,\beta=0}^{N_i-1}\left|H_{i,\alpha\beta}-H^{\mathrm{ref}}_{i,\alpha\beta}\right| + \sqrt{\frac{1}{N_\mathrm{e}}\sum_{i=1}^{N}\sum_{\alpha,\beta=0}^{N_i-1}(H_{i,\alpha\beta}-H^{\mathrm{ref}}_{i,\alpha\beta})^2},$$

where N is the number of structures, N_i is the dimension of the ith Hamiltonian, and N_e is the total number of matrix elements. Parameters were optimized by the AdamW [55,56] algorithm with an initial learning rate of 0.02. To accelerate convergence, we use the ReduceLROnPlateau method to schedule the learning rate: the learning rate will be reduced by a factor of 0.9 if accuracy is not improved within 50 epochs. We trained for 4000 epochs in graphene/h-BN and for 3000 epochs in the remaining systems. All models were trained on one NVIDIA 4090 GPU. More hyperparameters are provided in the SM.

Data and Code Availability. The training datasets used in this paper (multilayer MoS_2, graphene/h-BN,

and phosphorus allotropes) are available on Zenodo (https://doi.org/10.5281/zenodo.14807957). The HotHam code is available on Github (https://github.com/liangzhixin-202169/Hot-Ham).

Acknowledgements. This work was supported by the National Natural Science Foundation of China (Grant Nos. 12125404, T2495231, and 123B2049), the Basic Research Program of Jiangsu (Grant Nos. BK20233001, BK20241253, and BK20253009), the Jiangsu Funding Program for Excellent Postdoctoral Talent (Grant Nos. 2024ZB002 and 2024ZB075), the Postdoctoral Fellowship Program of CPSF (Grant No. GZC20240695), the AI & AI for Science program of Nanjing University, the Artificial Intelligence and Quantum physics (AIQ) program of Nanjing University, and the Fundamental Research Funds for the Central Universities. The calculations were carried out using supercomputers at the High Performance Computing Center of Collaborative Innovation Center of Advanced Microstructures, the high-performance supercomputing center of Nanjing University.

References

[1] Slater J C and Koster G F 1954 *Phys. Rev.* **94** 1498
[2] Wang Z, Ye S, Wang H, He J, Huang Q, and Chang S 2021 *npj. Comput. Mater.* **7** 11
[3] Gu Q, Zhang L, and Feng J 2022 *Sci. Bull.* **67** 29
[4] Gu Q, Zhouyin Z, Pandey S K, Zhang P, Zhang L, and E W 2024 *Nat. Commun.* **15** 6772
[5] Schütt K T, Gastegger M, Tkatchenko A, Müller K R, and Maurer R J 2019 *Nat. Commun.* **10** 5024
[6] Unke O T, Bogojeski M, Gastegger M, Geiger M, Smidt T, and Müller K R 2021 arXiv:2106.02347 [physics.chem-ph]
[7] Yu H, Xu Z, Qian X, Qian X and Ji S 2023 *Proceedings of the 40th International Conference on Machine Learning* (PMLR) **202** 40412
[8] Zhang L, Onat B, Dusson G, McSloy A, Anand G, Maurer R J, Ortner C, and Kermode J R 2022 *npj Comput. Mater.* **8** 158
[9] Nigam J, Willatt M J, and Ceriotti M 2022 *J. Chem. Phys.* **156** 014115
[10] Li H, Wang Z, Zou N, Ye M, Xu R, Gong X, Duan W, and Xu Y 2022 *Nat. Comput. Sci.* **2** 367
[11] Gong X, Li H, Zou N, Xu R, Duan W, and Xu Y 2023 *Nat. Commun.* **14** 2848
[12] Zhong Y, Yu H, Su M, Gong X, and Xiang H 2023 *npj Comput. Mater.* **9** 182
[13] Wang Y, Li H, Tang Z, Tao H, Wang Y, Yuan Z, Chen Z, Duan W and Xu Y 2024 arXiv:2401.17015 [physics.comp-ph]
[14] Zhouyin Z, Gan Z, Pandey S K, Zhang L and Gu Q 2024 arXiv:2407.06053 [cond-mat.mtrl-sci]
[15] Wang Y, Liang Z, Ding C, Wang J, Fan Z, Xing D and Sun J 2025 arXiv:2509.06525 [cond-mat.mtrl-sci]
[16] Scarselli F, Gori M, Tsoi A C, Hagenbuchner M, and Monfardini G 2009 *IEEE Transactions on Neural Networks* **20** 61
[17] Battaglia P W, Hamrick J B, Bapst V, Sanchez-Gonzalez A, Zambaldi V, Malinowski M, Tacchetti A, Raposo D, Santoro A, Faulkner R, Gulcehre C, Song F, Ballard A, Gilmer J, Dahl G, Vaswani A, Allen K, Nash C, Langston V, Dyer C, Heess N, Wierstra D, Kohli P, Botvinick M, Vinyals O, Li Y, and Pascanu R 2018 arXiv:1806.01261 [cs.LG]
[18] Kipf T N and Welling M 2017 arXiv:1609.02907 [cs.LG]
[19] Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, Wang L, Li C, and Sun M 2021 *AI Open* **1** 57
[20] Gilmer J, Schoenholz S, Riley P F, Vinyals O and Dahl G E 2017 *Proceedings of the 34th International Conference on Machine Learning* (PMLR) **70** 1263
[21] Passaro S and Zitnick C L 2023 *Proceedings of the 40th International Conference on Machine Learning* (PMLR) **202** 27420
[22] Batatia I, Kovács D P, Simm G N C, Ortner C, and Csányi G 2023 arXiv:2206.07697 [stat.ML]
[23] Xin H, Zhou Z, An D, Yan L Q, Xu K, Hu S M, and Yau S T 2021 *ACM Trans. Graph.* **40** 1
[24] Luo S, Chen T, and Krishnapriyan A S 2024 arXiv:2401.10216 [cs.LG]
[25] Blöchl P E 1994 *Phys. Rev. B* **50** 17953
[26] Kresse G and Furthmüller J 1996 *Phys. Rev. B* **54** 11169
[27] Kresse G and Furthmüller J 1996 *Comput. Mater. Sci.* **6** 15
[28] Ozaki T and Kino H 2004 *Phys. Rev. B* **69** 195113
[29] Ozaki T 2003 *Phys. Rev. B* **67** 155108
[30] Li P, Liu X, Chen M, Lin P, Ren X, Lin L, Yang C, and He L 2016 *Comput. Mater. Sci.* **112** 503
[31] Lin P, Ren X, Liu X, and He L 2024 *WIREs Comput. Mol. Sci.* **14** e1687
[32] Liao Y L and Smidt T 2023 arXiv:2206.11990 [cs.LG]
[33] Batzner S, Musaelian A, Sun L, Geiger M, Mailoa J P, Kornbluth M, Molinari N, Smidt T E, and Kozinsky B 2022 *Nat. Commun.* **13** 2453
[34] Musaelian A, Batzner S, Johansson A, Sun L, Owen C J, Kornbluth M, and Kozinsky B 2022 *Nat. Commun.* **14** 579
[35] Wigner E P and Fano U 1960 *Am. J. Phys.* **28** 408
[36] Xie Y, Daigavane A, Kotak M, and Smidt T 2025 *Proceedings of the 42nd International Conference on Machine Learning* (PMLR) **267** 68599
[37] Healy D M, Rockmore D N, Kostelec P J, and Moore S 2003 *J. Fourier Anal. Appl.* **9** 341
[38] Weiler M, Geiger M, Welling M, Boomsma W, and Cohen T 2018 arXiv:1807.02547 [cs.LG]
[39] Liao M, Wei Z, Du L J *et al.* 2020 *Nat. Commun.* **11** 2153
[40] Zhang S, Xu Q, Hou Y, Song A, Ma Y, Gao L, Zhu M, Ma T, Liu L, Feng X Q, and Li Q 2022 *Nat. Mater.* **21** 621
[41] Moon P and Koshino M 2014 *Phys. Rev. B* **90** 155406
[42] Wang E, Lu X, Ding S, Yao W, Yan M, Wan G, Deng K, Wang S, Chen G, Ma L, Jung J, Fedorov A V, Zhang Y, Zhang G, and Zhou S 2016 *Nat. Phys.* **12** 1111
[43] Wang J, Ma F C, Liang W J, and Sun M T 2017 *Mater. Today Phys.* **2** 6
[44] Wang V, Xu N, Liu J C, Tang G, and Geng W T 2021 *Comput. Phys. Commun.* **267** 108033
[45] Jain A, Ong S P, Hautier G, Chen W, Richards W D, Dacek S, Cholia S, Gunter D, Skinner D, Ceder G, and Persson K A 2013 *APL Materials* **1** 011002
[46] Zhang L, Huang H, Zhang B, Gu M, Zhao D, Zhao X, Li L, Zhou J, Wu K, Cheng Y, and Zhang J 2020 *Angew. Chem. Int. Ed.* **59** 1074
[47] Wang J, Gao H, Han Y, Ding C, Pan S, Wang Y, Jia Q, Wang H T, Xing D, and Sun J 2023 *Natl. Sci. Rev.* **10** nwad128
[48] Wang Y, Li Y, Tang Z, Li H, Yuan Z, Tao H, Zou N, Bao T, Liang X, Chen Z, Xu S, Bian C, Xu Z, Wang C, Si C, Duan W, and Xu Y 2024 *Sci. Bull.* **69** 2514
[49] Zhong Y, Yu H, Yang J, Guo X, Xiang H, and Gong X 2024 *Chin. Phys. Lett.* **41** 077103
[50] Zhong Y, Wang R, Gong X, and Xiang H 2025 arXiv:2504.19586 [cond-mat.mtrl-sci]
[51] Fan Z, Garcia J H, Cummings A W, Barrios-Vargas J E, Panhans M, Harju A, Ortmann F, and Roche S 2021 *Phys. Rep.* **903** 1
[52] Li Y, Zhan Z, Kuang X, Li Y, and Yuan S 2023 *Comput. Phys. Commun.* **285** 108632

[53] Perdew J P, Burke K, and Ernzerhof M 1996 *Phys. Rev. Lett.* **77** 3865
[54] Larsen A H, Mortensen J J, Blomqvist J, Castelli I E, Christensen R, Dułak M, Friis J, Groves M N, Hammer B, Hargus C, Hermes E D, Jennings P C, Jensen P B, Kermode J, Kitchin J R, Kolsbjerg E L, Kubal J, Kaasbjerg K, Lysgaard S, Maronsson J B, Maxson T, Olsen T, Pastewka L, Peterson A, Rostgaard C, Schiøtz J, Schütt O, Strange M, Thygesen K S, Vegge T, Vilhelmsen L, Walter M, Zeng Z, and Jacobsen K W 2017 *J. Phys.: Condens. Matter* **29** 273002
[55] Loshchilov I and Hutter F 2019 arXiv:1711.05101 [cs.LG]
[56] Reddi S J, Kale S and Kumar S 2019 arXiv:1904.09237 [cs.LG]