# Learning local equivariant representations for quantum operators

Zhanghao Zhouyin,[1,2] Zixi Gan,[1,3] Shishir Kumar Pandey,[4] Linfeng Zhang,[1,5] and Qiangqiang Gu[1,6,*]

[1]*AI for Science Institute, Beijing 100080, China*
[2]*Department of Physics, McGill University, Montreal, Quebec, Canada H3A2T8*
[3]*Department of Chemistry, Zhejiang University, Hangzhou 310027, China*
[4]*Birla Institute of Technology & Science, Pilani-Dubai Campus, Dubai 345055, UAE*
[5]*DP Technology, Beijing 100080, China*
[6]*School of Mathematical Science, Peking University, Beijing 100871, China*
(Dated: July 10, 2024)

Predicting quantum operator matrices such as Hamiltonian, overlap, and density matrices in the density functional theory (DFT) framework is crucial for understanding material properties. Current methods often focus on individual operators and struggle with efficiency and scalability for large systems. Here we introduce a novel deep learning model, SLEM (strictly localized equivariant message-passing) for predicting multiple quantum operators, that achieves state-of-the-art accuracy while dramatically improving computational efficiency. SLEM's key innovation is its strict locality-based design, constructing local, equivariant representations for quantum tensors while preserving physical symmetries. This enables complex many-body dependence without expanding the effective receptive field, leading to superior data efficiency and transferability. Using an innovative SO(2) convolution technique, SLEM reduces the computational complexity of high-order tensor products and is therefore capable of handling systems requiring the $f$ and $g$ orbitals in their basis sets. We demonstrate SLEM's capabilities across diverse 2D and 3D materials, achieving high accuracy even with limited training data. SLEM's design facilitates efficient parallelization, potentially extending DFT simulations to systems with device-level sizes, opening new possibilities for large-scale quantum simulations and high-throughput materials discovery.

## 1. Introduction

Quantum operators, representing observables and the evolution of quantum systems, are the cornerstone of describing the microscopic world. In modern quantum science, the advent of density functional theory (DFT) [1, 2] has elevated single-particle quantum operators, such as the Kohn-Sham Hamiltonian, density matrix, and overlap matrix, to paramount importance in solving complex problems [3]. These operators play a crucial role in unraveling electronic structures, predicting material properties, and advancing quantum technologies. However, as we tackle larger and more complex systems, the efficient and accurate representation of these fundamental operators has emerged as a pressing challenge in computation, demanding new avenues for innovative methodologies.

Recent advances have incorporated machine learning techniques to accelerate DFT calculations by directly predicting DFT's output of quantum operators, including charge density [4], overlapping matrix [4, 5], self-energy [6], wave function [4], and Hamiltonian matrix [4, 5, 7–10]. By circumventing self-consistent DFT calculations, such methods have the potential to scale up the electronic structure calculations. Some of the approaches utilize Gaussian regressions [9], kernel-based models [9], and neural networks [11] to predict invariant Hamiltonian matrix blocks on localized frames. Notably, powerful equivariant message-passing neural networks (E-MPNNs) have demonstrated remarkable accu-
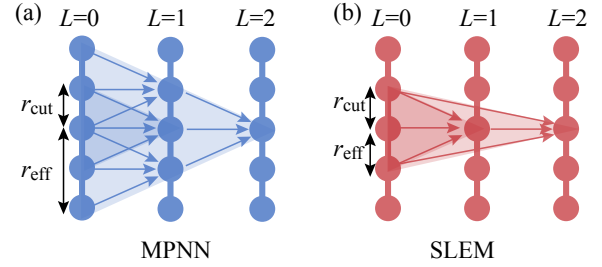


FIG. 1. Illustration of the local design of SLEM in comparison with conventional MPNN on a 1D graph. (a) MPNN information aggregation. (b) SLEM information aggregation. Balls and sticks represent nodes and edges; arrows show aggregation direction. $r_{\text{cut}}$: pre-defined cut-off; $r_{\text{eff}}$: effective cut-off after two hidden layer updates. $L$ is the layer index.

racy [12–20]. These networks ensure the equivariance of output tensor blocks, respecting the physical priors of atomic systems. Typically, they initialize the node and edge features with atomic species and the weighted projection of the spherical harmonics coefficient, then use iterative two-body updates to build many-body interactions. Despite their high accuracy, these iterative updates invariably enlarge the receptive field of node and edge features, incorporating numerous effective neighbours. This expansion limits parallelization and, consequently, the model's scalability. The storage-intensive nature of quantum tensor prediction tasks exacerbates these limitations, as high-dimensional features in edges and nodes necessitate substantial GPU memory during training and prediction. Consequently, training such

* guqq@aisi.ac.cn; guqq@pku.edu.cn

models on large datasets or predicting quantum operators for extensive atomic structures poses considerable challenges.

Fortunately, the electrostatic screening counteracts the long-term dependency in a lot of material systems [21–23]. For example, for bulk Silicon, Ge, a typical screening length is around 5 a.u. [23]. Therefore, the predicted quantum operators can be decomposed into elements with localized dependency on atomic structures. Naturally, a strictly local model that avoids expanding the receptive field, while maintaining sufficient accuracy is at most preferred. In the field of machine-learning interatomic potentials (MLIPs), this concept has been applied in the Allegro model [13], where the strictly local representation of interatomic potentials achieves high accuracy comparable to E-MPNNs and is highly parallelizable, particularly for large systems. However, unlike modelling interatomic potential, which only concerns scalars (energy) and vectors (forces) (corresponding to angular momentum $l = 0$ and 1) from node features, the prediction of the general representation of quantum operators necessitates targeting both node and edge features on high-order spherical tensors (even up to $l = 6$ or 8). This adds complexity to constructing strictly localized models for node and edge updates. Another significant challenge in high-order tensor prediction tasks is computational complexity. To equivariantly mix the features of different angular momentum $l$, tensor products that scale as $\mathcal{O}(l^6)$ are required [19]. Achieving high accuracy often necessitates up to hundreds of thousands of training steps, which significantly limits the applicability of current models to heavy elements. This limitation hinders the development of a unified machine-learning DFT model that generalizes across the periodic table. Therefore, a crucial question, whether one can employ the strictly localized scheme to develop powerful equivariant atom and bond features that can efficiently and accurately reproduce the corresponding quantum tensor block elements with reduced computational complexity, naturally arises.

This work presents a novel method, the strictly localized equivariant message-passing model (SLEM), implemented in the framework of DeePTB package [24] for efficient representations of quantum operators. SLEM employs a fully localized scheme to construct high-order node and edge equivariant features for a general representation of quantum operators, including the Hamiltonian and density matrix. As illustrated in Fig.1, the model embeds localized edge hidden states and utilizes them to construct localized node and edge features without including distant neighbours beyond a prefixed cutoff range $r_{\mathrm{cut}}$. This design enables the SLEM model to generalize more effectively, parallelize more easily, and scale to larger systems. Additionally, a fast and efficient SO(2) convolution [19] is implemented to reduce the $\mathcal{O}(l^6)$ complexity, with edge-specific training weights, thereby further enhancing the model's accuracy. As for the overlap matrix, it is typically required along with other operators

for property calculations. Previous works have used another E-MPNN that doubles the size of the networks [10] or extracts it from DFT calculations [5, 8], incurring additional costs or incorporating out-of-loop computation steps that complicate inference. In contrast, we utilized the two-center integrals nature and parameterized the overlap matrix with spherical independent scalars (i.e., Slater-Koster-like parameters [25]). This method, inspired by the early work of DeePTB [24], fits the overlap operator representation alongside other operators with minimal additional cost.

This paper is organized as follows. In section 2, we review the related works, focusing on message-passing neural networks and equivariant message passing. In section 3, we detail the architecture of the SLEM model. Section 4 presents our experimental results, comparing the accuracy and efficiency of our model with existing models in reported data sets, and conducting generalization experiments. Finally, we summarize our findings and discuss future directions.

## 2. Related works

*Message-passing Neural Networks* Message-passing neural networks (MPNNs) [26] have been widely applied in the modelling of atomic systems due to their exceptional accuracy in capturing the intricate relationships between atomic environments and physical properties [27]. In the message-passing scheme, atoms are treated as nodes in graphs, with bonds to neighbouring atoms represented as connected edges within a specified cutoff radius. The embedded atomic features are processed by trainable functions, generating messages from each edge to update the embeddings of central atoms. Formally, the MPNN framework can be summarized as follows:

$$\mathbf{e}^{ij,L} = \mathcal{N}_L \left( \mathbf{n}^{i,L-1}, \mathbf{n}^{j,L-1}, \mathbf{e}^{ij,L-1} \right)$$
$$\mathbf{m}^{ij,L} = M_L \left( \mathbf{n}^{i,L-1}, \mathbf{n}^{j,L-1}, \mathbf{e}^{ij,L} \right)$$
$$\mathbf{n}^{i,L} = U_L \left( \mathbf{n}^{i,L-1}, \sum_{j \in \mathcal{N}(i)} \mathbf{m}^{ij,L} \right)$$

Here, $\mathbf{e}^{ij,L}$ represents the edge features, $\mathbf{m}^{ij,L}$ denotes the messages, and $\mathbf{n}^{i,L}$ indicates the node features at layer $L$. $\mathcal{N}_L$, $M_L$, and $U_L$ are the trainable functions for the edge, message, and node updates. $\mathcal{N}(i) = \{j | r_{ij} < r_{\mathrm{cut}}\}$ indicates all the neighbour atoms for atom $i$, with $r_{\mathrm{cut}}$ being the predefined cutoff. This updating framework allows for the construction of many-body interactions and long-term dependencies, leading to strong performance across various applications. Previous works have predominantly utilized this updating scheme, achieving remarkably high precision in reported systems [12, 28, 29]. However, as updates progress, the effective cutoff radius ($r_{\mathrm{eff}} = N \times r_{\mathrm{cut}}$) for each atom's features grows linearly with the number of update steps ($N$), as shown in Fig.1.

Consequently, the effective neighbour list scales cubically with $r_{\text{eff}}$, making parallelization intractable. This issue is particularly severe when fitting representations of quantum operators, as the $r_{\text{cut}}$ defined by the cutoff radius for the linear combination of atomic orbitals (LCAO) basis is often larger than that defined in other tasks. A localized approach to constructing many-body interactions without including distant atoms has been proposed. For example, the Allegro model [13] modifies the updating rules by incorporating a hidden atom-pair state that depends explicitly on the direction vector and the center atom embedding. While this framework works successfully for potential energy prediction, it requires further improvement for fitting more general edge and node features. Other local methods avoid iterative updates that enlarge the receptive field, instead creating manually designed descriptors of the local environment [30–35]. These methods are generally local but are criticized for insufficiency in describing high-order interactions.

*Equivariant Message Passing*  There are certain natures of the physical quantities that should be invariant or equivariant under e.g. the spatial and temporal operations. For example, under translation, rotation, and reflection, the scalars such as the energy of a system are invariant, while the tensors, such as forces and Hamiltonian matrices should transform accordingly. Such three kinds of operations constitute the group operations of E(3), known as the Euclidian Group (O(3) if without translation). To model such equivariant quantities, a set of neural network models has been developed that incorporate operations that preserve this equivariance. [36–39] These neural networks possess physical priors, as their operations ensure the output transforms in sync with the input, making them more generalizable, accurate, and data-efficient in predicting physical quantities. Formally, an equivariant operation from vector space X to vector space Y is defined such that:

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

where $D_X[g] \in GL(X)$ is the representation of group element $g$ on vector space $X$. Since the representation takes the form of the direct sum of irreducible representations of the group $G$, it suggests that the vectors $x$ and $y$ themselves are composed by the irreducible representation (irreps for short) of group $G$. Usually, we consider here for operations of group O(3)(since translational symmetry is naturally satisfied by using relative coordinates), the irreps are the spherical tensors with angular momentum index $l$ and parity $p$. So features $x, y$ should also be indexed by $l, p$. For each irrep with angular momentum $l$, an internal degree of freedom, denoted by $m$ exists such that $|m| \leq l$. Therefore, each irreps of order $l$ belongs to a subspace of dimension $2l+1$. Once the $l, p$ index is defined, one can construct the group operations, which are represented by the $(2l+1) \times (2l+1)$ Wigner-D matrices.

Performing numerical operations on such irreps features follows certain rules. For the tensors with the same $l$, addition and subtraction can be conducted. To mix up the tensors of different $l$ indices, an important operation called tensor product ($\otimes$) is applied, which is considered the generalized operation of multiplication on spherical tensors. More formally:

$$(\mathbf{x} \otimes \mathbf{y})_{m_3}^{l_3} = \sum_{m_1, m_2} C_{(l_1,m_1)(l_2,m_2)}^{(l_3,m_3)} \mathbf{x}_{m_1}^{l_1} \mathbf{y}_{m_2}^{l_2}$$

Where $C_{(l_1,m_1)(l_2,m_2)}^{(l_3,m_3)}$ are Clebsch-Gordan (CG) coefficients. This operation is found extremely useful in constructing edge and node updating functions, as seen in []. However, the conventional tensor product has the time and memory scales of $O(l_{max}^6)$ [19] where $l_{max}$ is the maximum angular momentum of $\mathbf{x}$ and $\mathbf{y}$. Such complexity limits the application like the equivariant forces fields where high inference speed is vital. It poses an even greater challenge for quantum tensor prediction tasks, which require tensor products on very high-order spherical tensors. For example, constructing blocks of *d-d*, *f-f* and *g-g* orbital pairs require irreps of maximum order $l = 4$, $l = 6$ and $l = 8$, respectively. Such high costs make training for large-size systems nearly impossible.

## 3.  Model Architecure

### A.  Parameterize Equivariant Quantum Operators

The quantum operators $\hat{O}$, such as the Hamiltonian and density matrix in the LCAO-based DFT framework, follow the equivariance of the $E(3)$ group and can thus be expressed in tensorized features using group theory. This process is illustrated in Fig. 2. Specifically, the $E(3)$ group consists of spatial rotation, translation, and reflection. Translational symmetry is automatically satisfied by performing operations only on relative coordinates, while rotation and reflection require special treatments. Under the LCAO basis, the quantum tensor block can be expressed as:

$$O_{l_1,l_2,m_1,m_2}^{i,j} = \langle i, l_1, m_1 | \hat{O} | j, l_2, m_2 \rangle$$

Here $i$ and $j$ denote atomic sites, while $l_1, m_1$, and $l_2, m_2$ refer to the atomic orbitals of site $i$ and site $j$ respectively. The terms $m_1$ and $m_2$ are the magnetic quantum numbers associated with the angular momentum $l_1$ and $l_2$. $O_{l_1,l_2,m_1,m_2}^{i,j}$ are the matrix element of operator $\hat{O}$. When the coordinates of the system are transformed by a rotation operation $\mathbf{R}$, equivariance requires the quantum tensor to transform accordingly as:

$$O_{l_1,l_2,m_1',m_2'}^{\prime i,j} = \sum_{m_1, m_2} D(\mathbf{R})_{m_1, m_1'}^{l_1} D^*(\mathbf{R})_{m_2, m_2'}^{l_2} O_{l_1,l_2,m_1,m_2}^{i,j} \tag{1}$$

Where $O'$ is the equivariant block after transformation, and $D(\mathbf{R})_{m,m'}^l$ is the Wigner-D matrix of order $l$. To con-
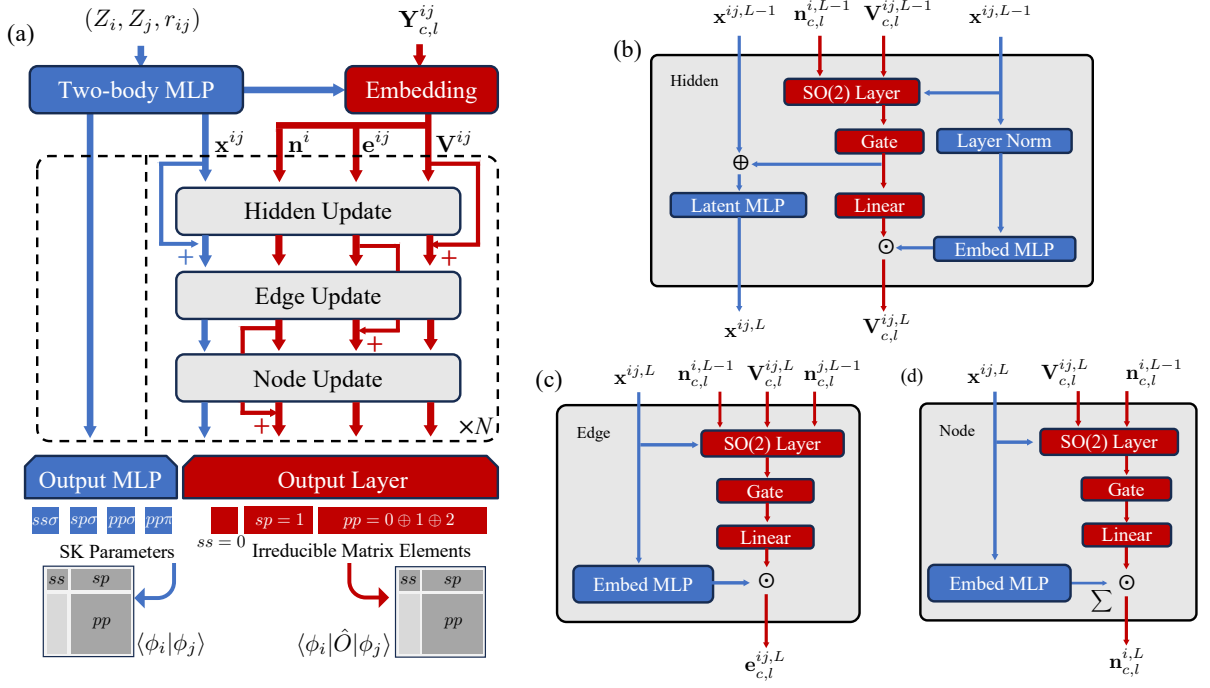
FIG. 2. The design of the SLEM model. (a) the hierarchy structure of the model. Starts with the structure information such as atomic number $Z_i$, and the radial and spherical part of the shift vector $r_{ij}$ and $\mathbf{Y}_{c,l}^{ij}$, the initialized hidden $\mathbf{x}^{ij}$, $\mathbf{V}^{ij}$, edge and node features $\mathbf{e}^{ij}$, $\mathbf{n}^i$ would be generated. The initialized two-body hidden features would be used to predict the Slater-Koster-like Overlap parameters directly and transformed to construct (off-)diagonal blocks for the overlap operator. Others then, will update iteratively using the design's strict localized updating scheme. (b)-(d) shows the details of the hidden update (b), edge update (c), and node update (d). After the updates, the node and edge features would be used to construct the (off-)diagonal blocks using the Wigner-Eckart theorem for quantum operators such as density matrix and Hamiltonian.

struct such equivariance blocks from irreps, the Wigner-Eckart theorem can be applied, which decomposes the operator indexed by $l_1, l_2$ into a single index $l_3$ that satisfies $|l_1 - l_2| \leq l_3 \leq (l_1 + l_2)$, which states:

$$o_{l_3,m_3}^{i,j} = \sum_{l_1,m_1,l_2,m_2} C_{(l_1,m_1)(l_2,m_2)}^{(l_3,m_3)} O_{l_1,l_2,m_1,m_2}^{i,j} \quad (2)$$

Here, the edge $(i \neq j)$ and node $(i = j)$ features $o_{l_3,m_3}^{i,j}$ are grouped by the index $m$ into vectors of $\mathbf{o}_{c,l}^{i,j}$ with $c$ accounting for multiple tensors for the same $l$. These features can be computed for hopping $(i \neq j)$ and onsite $(i = j)$ elements of the quantum operators. Further by leveraging the Hermitian nature of quantum operators, the parameterized elements can be further reduced to upper diagonal blocks.

The decomposed features $\mathbf{o}_{c,l}^{i,j}$ are aligned with the network prediction through a rescaling procedure. The target node/edge equivariant features $\mathbf{o}_{c,l}^{i,i/j}$ can be standarized and decomposed into constant norms and biases, and the normalized features that have balanced variance. Formally:

$$\mathbf{o}_{c,l}^{i,i/j} = \sigma_{c,l}^{Z_i,Z_{i/j}} \hat{\mathbf{o}}_{c,l}^{i,i/j} + \mu_{c,l}^{Z_i,Z_{i/j}} \delta_{l,0}$$

Here, $Z_i$, $Z_j$ are the atom species of atom $i$ and $j$, $\sigma_{c,l}^{Z_i,Z_{i/j}}$ and $\mu_{c,l}^{Z_i,Z_{i/j}}$ are norm and bias value for each atom and atom-pair. These values are derived from the dataset statistics and subsequently applied as weights and shift biases in an atom/bond type-specific scaling layer. The improvement of the standardization procedure on enhancing model convergence is evident. For the tensor operators in LCAO basis, the diagonal blocks (or node features) often exhibit highly unbalanced norms across irreps of different $l$, hindering the model from learning a smooth function. Additionally, different irreps in the off-diagonal blocks (or edge features) demonstrate diverse decaying behaviors, making it challenging to learn these decays implicitly. By normalizing in this manner, the decay function for each irrep is constrained to a value ranging from 1 to 0. This facilitates the use of hidden states and a few layers of a Multilayer Perceptron (MLP) activated by a ReLU-like function, to explicitly project the decaying value from the preserved hidden scalar features for each edge. Detailed analysis of the decaying behavior of edge irreps and the norm of node irreps can be found in the supplementary materials (SM).

At this stage, the objective is to generate the node and edge features $\mathbf{n}_{c,l}^i$, $\mathbf{e}_{c,l}^{ij}$, and to supervise them towards achieving decomposed target features $\hat{\mathbf{o}}_{c,l}^{i,i}$ and $\hat{\mathbf{o}}_{c,l}^{i,j}$. Once a satisfactory accuracy is attained, the inversion of the Wigner-Eckart theorem facilitates the reconstruction of predicted operator representations such as Hamiltonian

and density matrix blocks:

$$O^{i,j}_{l_1,l_2,m_1,m_2} = \sum_{l_3,m_3} C^{(l_3,m_3)}_{(l_1,m_1)(l_2,m_2)} o^{i,j}_{l_3,m_3}$$

The whole procedure satisfies the rotational, transformational and reflectional symmetry.

## B. Parameterize of Invariant Overlap Operators

The overlap matrix, analogous to other quantum operators, is defined as:

$$S^{i,j}_{l_1,l_2,m_1,m_2} = \langle i, l_1, m_1 | j, l_2, m_2 \rangle$$
$$= \int \phi^{i,l_1}_{m_1}(\mathbf{r}) \phi^{j,l_2}_{m_2}(\mathbf{r} - \mathbf{r}_{ij}) d\mathbf{r}$$

where $\phi^{i,l}_m$ is the LCAO basis. The overlap matrix also satisfies the equivariance relation defined in Eq.1. Since the equivariant dependency comes fully from the bases, it is possible to rotate them to align with the axis of $\mathbf{r}_{ij}$ to reduce the angular dependence, simplifying the matrix elements further into scalars via the relation [40]:

$$\int \phi^{i,l_1}_{m_1}(\mathbf{r}) \phi^{j,l_2}_{m_2}(\mathbf{r} - \mathbf{r}_{ij}) d\mathbf{r} = s^{Z_i,Z_j}_{l_1,l_2,|m_1|}(r_{ij}) \delta_{m_1,m_2}$$

Here, the dependence changes to $Z_i$, $Z_j$, and $r_{ij}$ due to the two-center nature of the overlap integrals. Parameterizing these distance-dependent radial functions involves encoding atomic species with radial information and employing a simple MLP to learn the mapping to target scalars.

$$f_{l_1,l_2,|m|}(Z_i, Z_j, r_{ij}) = s^{Z_i,Z_j}_{l_1,l_2,|m_1|}(r_{ij}) \delta_{m_1,m_2} \quad (3)$$

After getting these parameters, we use them to construct the overlap matrix aligned with the bond axis, and then rotate it back to the original orientation.

## C. The SLEM model

The SLEM model architecture is illustrated in Fig.2. The model maintains a set of features, including hidden features $\mathbf{x}^{ij,L}$, $\mathbf{V}^{ij,L}_{c,l}$, node features $\mathbf{n}^{i,L}_{c,l}$ and edge features $\mathbf{e}^{ij,L}_{c,l}$ of hidden layer $(L)$. Specifically, the hidden features consist of a scalar channel $\mathbf{x}^{ij,L}$ and a tensor channel $\mathbf{V}^{ij,L}_{c,l}$. The scalar channel is initialized as an embedded vector containing two-body information, including the atomic species and the radial distance of the atom pair. These initialized two-body radial features are then processed through an MLP to the invariant Slater-Koster-like parameters of overlap. For equivariant quantum operators, such as the Hamiltonian and density matrix, the scalar and tensor channels interact to generate an ordered atom-pair representation, which is used to construct local node and edge representations. After iterative updates, the representation is scaled by the statistical norm and biases, thereby achieving the final prediction.

### 1. Feature initialization

Firstly, the initial scalar hidden feature is computed from the two-body embeddings of atomic species $Z_i$ and $Z_j$, and the radial distance, as follows:

$$\mathbf{x}^{ij,L=0} = \mathbf{MLP}_{2\text{-body}}\left(\mathbf{1}(Z_i)||\mathbf{1}(Z_j)||\mathbf{B}(r_{ij})\right) \cdot u(r_{ij})$$

Here, atom species are embedded with one-hot vectors denoted as $\mathbf{1}(Z)$, and a set of trainable Bessel bases $\mathbf{B}(r_{ij})$ is utilized to encode the distance $r_{ij}$ between atoms $i$ and $j$. $u(r_{ij})$ are envelope functions [41] to add explicit radial dependence. Subsequently, the edge and hidden features are initialized as weighted spherical harmonics of relative edge vectors:

$$\mathbf{V}^{ij,L=0}_{c,l} = w_{c,l}\left(L_N\left(\mathbf{x}^{ij,L=0}\right)\right)\mathbf{Y}^{ij}_l$$
$$\mathbf{e}^{ij,L=0}_{c,l} = w_{c,l}\left(\mathbf{x}^{ij,L=0}\right)\mathbf{Y}^{ij}_l$$

Here, the weights are learned from the initialized scalar hidden features $\mathbf{x}^{ij,L=0}$. Layer normalization $L_N$ ensures that the hidden tensor features have a balanced amplitude of each edge. The initial node features are then computed as linear transformations of the aggregated edge features:

$$\mathbf{n}^{i,L=0}_{c,l} = \mathbf{Linear}\left(\frac{1}{\sqrt{N_{avg}}}\sum_{j\in\mathcal{N}(i)}\mathbf{e}^{ij,L=0}_{c,l}\right)$$

Here $\mathcal{N}(i)$ and $N_{avg}$ are the neighbouring atoms and the average number of neighbours of atom$-i$.

### 2. Speed up tensor product

To integrate the information from the equivariant features, the tensor product is employed in all updating blocks of the SLEM model. Generally, the tensor product in SLEM is performed with the concatenated equivariant features $\tilde{\mathbf{f}}^{ij}_{c,l}$ and the weighted projection of the edge shift vector $\boldsymbol{r}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j$ on the spherical harmonics function $\mathbf{Y}^{ij}_l$. Formally:

$$\mathbf{f}^{ij}_{c_3,l_3} = \tilde{\mathbf{f}}^{ij}_{c_1,l_1} \otimes w^{ij}_{c_2,l_2} \mathbf{Y}^{ij}_{l_2}$$
$$= \sum_{c_1,l_1,l_2} \tilde{w}^{ij}_{c_1,l_1,l_2} \sum_{m_1,m_2} C^{(l_3,m_3)}_{(l_1,m_1)(l_2,m_2)}$$
$$\cdot f^{ij}_{c_1,l_1,m_1} Y^{ij}_{l_2,m_2} \quad (4)$$

Here, $\tilde{w}^{ij}_{c_1,l_1,l_2} = \sum_{c_2} w_{c_1,c_2,l_1,l_2} w^{ij}_{c_2,l_2}$ are edge-specific parameters for each tensor product operation. Performing such tensor products on high-order features is computationally intensive. Therefore, we applied the recently developed SO(2) [19] convolution to simplify, reducing the computation and storage complexity from $O(l^6_{max})$ to $O(l^3_{max})$. The simplification idea is intuitive. $\mathbf{Y}^{ij}_{l_2,m_2}$ are

sparse tensors if rotated to align with the edge $ij$, which is nonzero only for $m_2 = 0$. Therefore, it is easier to compute the tensor production in the direction of edge $ij$, and rotate inversely the output afterwards. This step removes the $m_2$ index from the summation in Eq. 4. Additionally, considering Clebsch-Gordan coefficients with $m_2 = 0$, we find that $C_{(l_1,m_1)(l_2,0)}^{(l_3,m_3)} = 0$ except for $m_3 = \pm m_1$. This allows further reduction of the summation in Eq. 4 by replacing $\pm m_1$ with a single index $m$. Then the operations can be reformulated formally as:

$$
\begin{pmatrix} f_{c,l,m}^{ij} \\ f_{c,l,-m}^{ij} \end{pmatrix} = \sum_{c',l'} \begin{pmatrix} w_{c,c',l',m} - w_{c,c',l',-m} \\ w_{c,c',l',-m} \quad w_{c,c',l',m} \end{pmatrix} \cdot \begin{pmatrix} \tilde{f}_{c',l',m}^{ij} \\ \tilde{f}_{c',l',-m}^{ij} \end{pmatrix}
$$

This represents a linear operation on $\tilde{f}_{c',l',m'}^{ij}$. By employing this method, high-order tensor products for $l = 8, 9$ and even 10 can be efficiently calculated, which is essential for heavy element systems where the $f$, $g$ or even higher orbitals are used in the LCAO basis for DFT calculations. Finally, the weights for the new SO(2) tensor product method are multiplied by edge-specific parameters, where $w_{c',l'}^{ij}$ are mapped by an MLP from hidden scalar features $\mathbf{x}^{ij,L}$ as $\tilde{w}_{c,c',l',m}^{ij} = w_{c,c',l',m} w_{c',l'}^{ij}$. This powerful and efficient tensor product layer facilitates the construction of local interactive updates of the features.

### 3. Hidden updates

To construct many-body interactions, as in Fig. 2(b), the node features $\mathbf{n}_{c,l}^i$ and hidden tensor features $\mathbf{V}_{c,l}^{ij}$ would be concatenated and doing tensor product with the projection coefficients of edge shift vector $\boldsymbol{r}_{ij}$ on the spherical harmonics functions. The operation is written formally as:

$$
\tilde{\mathbf{V}}_{c_3,l_3}^{ij,L} = \left(\mathbf{n}^{i,L-1} || \mathbf{V}^{ij,L-1}\right)_{c_1,l_1} \otimes w_{c_2,l_2}^{ij,L} \mathbf{Y}_{l_2}^{ij} \qquad (5)
$$

Unlike most MPNN, the hidden states $\mathbf{x}^{ij}$ and $\mathbf{V}_{c,l}^{ij}$ in SLEM is only a function of the local environment of centre atom $i$, the shift vector of edge $ij$, and the information of atom $j$. Such a design excludes neighbours of $j$ into hidden states.

After the tensor production, the output features $\tilde{\mathbf{V}}_{c,l}^{ij,L}$ will be passed through the gated non-linearity [41], and transformed by an E3linear [42] layer to mix up the information across different channels. The new hidden feature will be multiplied by the weights learned from normalized scalar features to include the radial information explicitly and return as the updated feature $\mathbf{V}_{c,l}^{ij,L}$.

The scalar hidden features are updated by mixing the 0th order information from $\mathbf{V}_{c,l}^{ij,L}$ with a latent MLP, which is:

$$
\mathbf{x}^{ij,L} = \mathbf{MLP}\left(\mathbf{x}^{ij,L-1} || \mathbf{V}_{c,l=0}^{ij,L}\right) \cdot u(r_{ij}) \qquad (6)
$$

Containing the explicit decaying envelope function $u(r_{ij})$, and the many-body interactions of scalar and tensor features, the sclar hiddien states $\mathbf{x}^{ij,L}$ can be very useful to describing the explicit decaying behaviour of each edge irreps feature along the radial distance.

### 4. Node updates

The strict local node representation $\mathbf{n}_{c,l}^{ij,L}$ can be constructed naturally from the many-body interactive tensor features $\mathbf{V}_{c,l}^{ij,L}$. We follows the MPNN style to create the message from node-$j$ to node-$i$. Formally:

$$
\mathbf{m}_{c_3,l_3}^{ij,L} = \left(\mathbf{n}^{i,L-1} || \mathbf{V}^{ij,L}\right)_{c_1,l_1} \otimes w_{c_2,l_2}^{ij,L} \mathbf{Y}_{l_2}^{ij} \qquad (7)
$$

Here again, we exclude the neighbouring information of atom-$j$ in $\mathbf{m}_{c_3,l_3}^{ij,L}$ via this partial updates of $\mathbf{V}_{c,l}^{ij,L}$. However, all necessary interactions are contained in this format, including the interactions of atom-$j$, bond $i - j$, atom-$i$, and the environment of $i$.

Each message then is passed through a gated activation and E3Linear layer, and weighted separately by weights learnt from the hidden scalar features, by:

$$
\mathbf{n}_{c_3,l_3}^{ij,L} = \frac{1}{N_{avg}} \sum_{j \in \mathcal{N}(i)} w_{c_3,l_3}^{ij} \mathbf{m}_{c_3,l_3}^{ij,L} \qquad (8)
$$

The weights here are different from those in hidden updates since it is learnt from $\mathbf{x}^{ij,L}$ directly without passing through a normalization. Therefore, the absolute radial decay is enforced in the weights, giving strong priors that the message from a closer atom is often more important while maintaining flexibility. Meanwhile, combining the updates of hidden scalar $\mathbf{x}^{ij,L}$, we can see the weights $w_{c_3,l_3}^{ij}$ are dependent on the features $\mathbf{V}^{ij,L}$ and $\mathbf{n}^{i,L-1}$, as shown in Eq.6. Such style aligns surprisingly with the equivariant graph attention mechanism [16, 17] that shows powerful expressibility in various tasks. Here $w_{c_3,l_3}^{ij}$ corresponding to an attention score computed from $\mathbf{V}^{ij,L}$ and $\mathbf{n}^{i,L-1}$. Therefore, through this update, the node features' dependencies are kept strictly local.

### 5. Edge updates

The locality of edge features $\mathbf{e}_{c,l}^{ij,L}$ can be naturally preserved as long as the node features are localized strictly. By mixing the information of node features on both sides, localized updates can be formulated as:

$$
\tilde{\mathbf{e}}_{c_3,l_3}^{ij,L} = \left(\mathbf{n}^{i,L-1} || \mathbf{V}^{ij,L} || \mathbf{n}^{j,L-1}\right)_{c_1,l_1} \otimes w_{c_2,l_2}^{ij,L} \mathbf{Y}_{l_2}^{ij}
$$

Similarly, the updated edge features are processed via a gated activation and an E3Linear layer, and then multiplied with weights learnt from the hidden scalar features (without normalization) as:

$$
\mathbf{e}_{c_3,l_3}^{ij,L} = w_{c_3,l_3}^{ij} \tilde{\mathbf{e}}_{c_3,l_3}^{ij,L}
$$

TABLE I. MAE (in meV) for Hamiltonian matrix predictions using SLEM and other methods on materials with DFT-LCAO basis up to $d$, $f$, and $g$ orbitals. Numbers in parentheses indicate parameter count. Data for $HfO_2$ in DeePH-E3 and for both GaN and $HfO_2$ in HamGNN are absent due to out-of-memory errors.

| Material | SLEM | | DeePH-E3 | | HamGNN |
|---|---|---|---|---|---|
| | (0.7M) | (4.5M) | ( 1.0 M) | ( 4.5M) | (10M) |
| $MoS_2$ | **0.34** | **0.14** | 0.46 | 0.55 | 0.63 |
| Graphene | **0.26** | **0.14** | 0.40 | 0.28 | 0.40 |
| Si(300K) | **0.10** | **0.07** | 0.16 | 0.10 | 9.0 |
| Si(600K) | **0.20** | **0.13** | 0.19 | 0.14 | 9.7 |

Systems with LCAO-basis up to $d$-orbitals (above)

| | SLEM (1.7M) | DeePH-E3 (1.9M) |
|---|---|---|
| GaN | **0.21** | 0.87 |
| $HfO_2$ | **0.28** | - |

Systems with LCAO-basis up to $f$ and $g$-orbitals (above)

TABLE II. MAE for predicting density matrix using SLEM on materials with DFT-LCAO basis up to $d$, $f$, and $g$ orbitals. Model settings align with those in Table I.

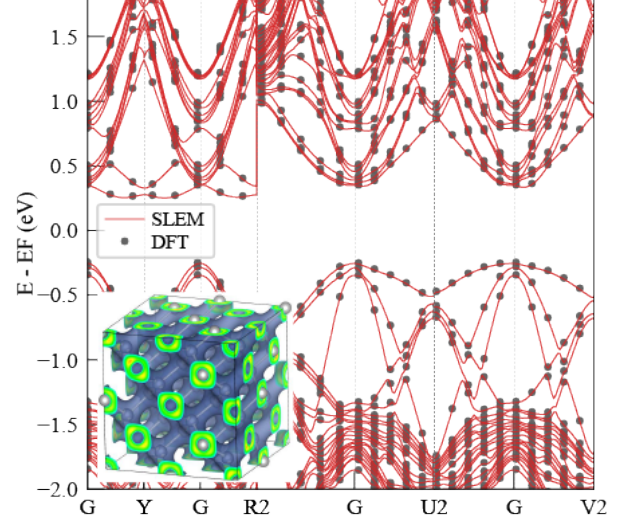| SLEM charge density model | | | |
|---|---|---|---|
| Materials | Silicon | GaN | $HfO_2$ |
| MAE | 8.9e-5 | 2.3e-5 | 3.9e-5 |



FIG. 3. Comparison of band structures for a Si MD trajectory snapshot: SLEM prediction vs. DFT calculation. Predicted band structures are obtained from either diagonalization of the predicted Hamiltonian or NSCF DFT calculation using predicted charge density, yielding indistinguishable results. Inset: Visualization of charge density distribution for the same structure.

Here we reframe the updating rules of SLEM with the MPNN framework, which looks like this:

$$\mathbf{V}^{ij,L} = \mathcal{V}_L \left( \mathbf{n}^{i,L-1}, \mathbf{V}^{ij,L-1} \right)$$
$$\mathbf{e}^{ij,L} = \mathcal{N}_L \left( \mathbf{n}^{i,L-1}, \mathbf{V}^{ij,L}, \mathbf{n}^{j,L-1} \right)$$
$$\mathbf{m}^{ij,L} = M_L \left( \mathbf{n}^{i,L-1}, \mathbf{V}^{ij,L} \right)$$
$$\mathbf{n}^{i,L} = U_L \left( \mathbf{n}^{i,L-1}, \sum_{j \in \mathcal{N}(i)} \mathbf{m}^{ij,L} \right)$$

Where $\mathcal{V}_L$ is the neural network for hidden feature construction. This update scheme constructs many-body interactions to build equivariant edge and node features while preserving the absolute locality by excluding atoms outside the constant cutoff radius. Such a scheme, in principle, would have much better transferability, data-efficiency, as well as scalability. Most importantly, it is possible to parallelize such update computations, allowing for training and inference on multiple devices. This is also vital for inferencing on very large atomic structures. The following section will focus on validating the effectiveness of this framework via learning equivariant DFT Hamiltonians and density matrices and overlap, which have very high-order tensors and need extremely high prediction accuracy. We demonstrate our method is much better than the previously reported ones in training and inference speed, transferability and the potential power to parralize the inference on multi-GPU devices.

## 4. Results

### A. Benchmark the accuracy and data-efficiency

We benchmark our model's performance in fitting Hamiltonian, density matrix, and overlap matrix using a diverse dataset. For the Hamiltonian, the datasets include systems with up to $d$ orbitals, encompassing 2D systems like single-layer $MoS_2$ and graphene from reported datasets [11], as well as 3D bulk silicon generated in this work. Given SLEM's proficiency with high-order spherical tensors, we also train it on our generated datasets of bulk GaN and $HfO_2$ systems, which include $f$ and $g$ orbitals, respectively. Structures in the Si, GaN, and $HfO_2$ systems are sampled via molecular dynamics using neural network potentials [34]. For density matrix and overlap matrix evaluations, we test the SLEM model exclusively on the datasets of Si, GaN, and $HfO_2$, as the reported datasets for $MoS_2$ and graphene lack density and overlap matrix data.

As shown in Table I, our SLEM model achieves state-

TABLE III. MAE for predicting overlap matrix using SLEM's parameterization on materials with DFT-LCAO basis up to $d$, $f$, and $g$ orbitals. Model settings align with those in Table I.

| SLEM overlap prediction | | | |
|---|---|---|---|
| Materials | Silicon | GaN | $HfO_2$ |
| MAE | 5.6e-5 | 4.7e-5 | 6.3e-5 |

TABLE IV. Comparison of validation MAE (in meV) for SLEM model and literature methods trained on randomly split datasets [11] with varying training ratios. Model settings align with those in Table I.

| MoS2 | | | | | |
|---|---|---|---|---|---|
| Partition | 100% | 80% | 60% | 40% | 20% |
| SLEM | **0.34** | **0.37** | **0.39** | **0.37** | **0.37** |
| DeePH-E3 | 0.46 | 0.72 | 0.84 | 1.03 | 1.46 |
| HamGNN | 1.11 | 1.15 | 1.38 | 1.65 | 1.68 |
| Graphene | | | | | |
| Partition | 100% | 80% | 60% | 40% | 20% |
| SLEM | **0.26** | **0.26** | **0.27** | **0.21** | **0.26** |
| DeePH-E3 | 0.40 | 0.30 | 0.33 | 0.36 | 0.60 |
| HamGNN | 0.41 | 0.32 | 0.37 | 0.42 | 0.60 |

of-the-art accuracy in Hamiltonian prediction, exhibiting the lowest mean absolute error (MAE) across all test systems compared to referenced methods. Notably, high accuracy is achieved with a relatively small model size (only 0.7 million (M) trainable parameters). Fig. 3 illustrates the band structure of silicon structures computed directly from the SLEM-predicted Hamiltonian, where the eigenvalues are indistinguishable from those obtained using DFT. For the density matrix, fitting results are presented in Table II. The results demonstrate very high accuracy (order of 1e-5), approaching the machine precision limit of float32 numbers. In Fig. 3, we use the trained model to predict the density matrix and visualize its real-space distribution. This capability is particularly important for applications such as charge distribution analysis or tracking electron transfer via methods like differential charge density. Furthermore, the predicted density can be directly used for non-self-consistent field (NSCF) DFT calculations. The resulting band structure for silicon, as an example, is highly accurate and matches the DFT output, with a MAE of only 1.09 meV in eigenvalues compared to self-consistent DFT results. The overlap matrix, represented by invariant parameters in our SLEM model, achieves exceptionally high accuracy as demonstrated in Table III, approaching the machine precision limit for float32 numbers. Notably, our simplified parameterization enables this high accuracy with only a minimal increase in model complexity. For instance, in a silicon model designed to fit only the Hamiltonian, our typical parameter count is 0.7 M. The inclusion of overlap matrix prediction adds merely 0.01 M parameters, representing an increase of only about 1.4% in total model size.

Additionally, due to the strict localization scheme, our SLEM model demonstrates superior data efficiency. Specifically, training the SLEM model requires fewer data points generated via DFT calculations. To quantify the data-efficiency performance, we designed an experiment by randomly splitting the dataset into subsets containing 20%, 40%, 60%, and 80% of the original training data. We then trained the strictly local SLEM model and other methods using these subsets and validated their performance on the same validation sets. The results, presented in Table IV, illustrate the remarkable data-efficiency of our method. Unlike the other methods, which showed a considerable decrease in accuracy with reduced data, our model maintained high accuracy across all subsets. This indicates that using SLEM allows users to generate a smaller, cost-effective training set. Furthermore,
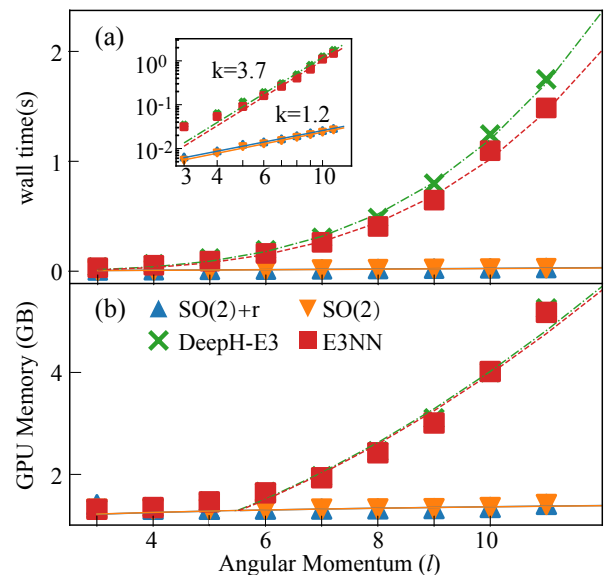


FIG. 4. Comparison of time and memory consumption for different tensor-product implementations. (a) Time consumption vs. angular momentum ($l$) for different models, including the SO(2)-based SLEM model (triangles) with and without radial part ($r$), DeepH-E3 (cross) [8], and E3NN (square) [42] models. Inset: Log-scale fit with slopes of 1.2 for the SLEM model and 3.7 for the other two models. (b) Memory consumption vs. $l$. The SLEM model demonstrates over two orders of magnitude improvement in both time and memory efficiency compared to DeepH-E3 and E3NN.

the excellent data efficiency and transferability of SLEM make it particularly suitable as a backbone model for developing a universal DFT model, especially for systems involving heavy elements.

## B.  Efficiency and Scalability

Many interesting and important properties in material science, chemistry and biology appear with systems composed of heavy atoms. The inclusion of heavy atoms would introduce very high-order spherical tensors when representing quantum operators such as the DFT Hamiltonian. Therefore, scaling to such systems would be quite a challenge because the tensor product used to construct complex spherical tensors scales as $O(L^6)$. It is very hard to train and inference on systems with heavy atoms using conventional tensor production. Moreover, inferring large material systems while training with small structures is particularly valuable. It therefore requires parallelising the model inference by assigning partitions of the whole atomic structure to multiple GPU workers. However, this is very difficult for most of the currently available models. Since the increasing receptive fields from the iterative updates of the graph would increase the minimal size of each partitioned subgraph, which makes such partition less effective. The SLEM model addresses these challenges by efficiently constructing high-order tensor products and assisting parallelization through its strict locality.

Firstly, for efficiency, the implementation of SO(2) convolution reduces the tensor product computational complexity from $O(L^6)$ to $O(L^3)$, which is then further reduced by the parallelization of matrix operations to near $O(L)$ benefiting from PyTorch. In Fig. 4, we compare the wall time and GPU memory consumed by tensor product operations using SO(2) convolution and the normal method employed in DeepH-E3 [8] and E3NN [42]. Clearly, tensor production via SO(2) convolution is far more efficient than conventional methods. This reduction of cost makes our method very efficient, which is capable of handling all possible basis choices in LCAO DFT. We also record the memory and time required when training on some of the typical systems. The results are displayed in Figure.5. We can see, that in all systems, our model trains faster in time while requiring less memory. Notably, the larger the basis set, the greater the advantage our method has in both memory and time efficiency.

Additionally, the strict local design assists the parallelization greatly. Since the edge and node features in the SLEM model are constructed locally, each node and edge feature is only dependent on the neighbour subgraph of the whole system, as:

$$\mathbf{n}^{i,L} = U_L \left( \mathbf{n}^{i,L-1}, \sum_{j \in \mathcal{N}(i)} \mathbf{m}^{i,L} \right)$$
$$= U(Z_i, \{\mathbf{Y}^{ij}, Z_j\}_{j \in N(i)}) \qquad (9)$$

Therefore, it is possible to divide the whole atomic system into sub-graphs and compute their node and edge features independently on different devices. This is extremely important when expanding the simulation power of DFT to very large systems. In practice, we find that for
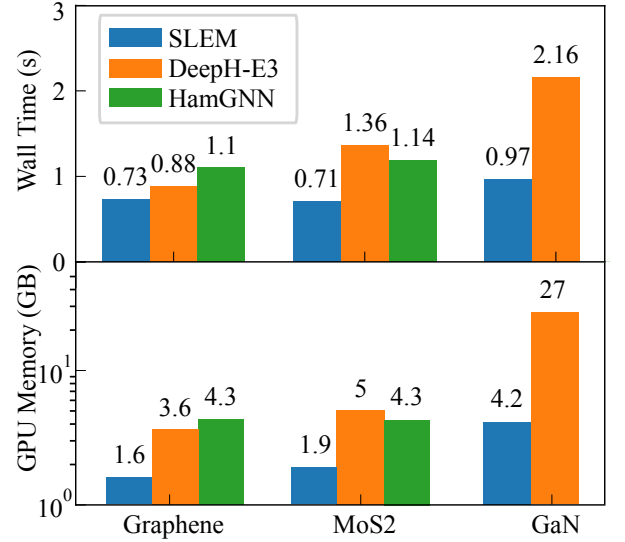


FIG. 5. Comparison of training time per iteration and memory consumption among SLEM, DeepH-E3 [8], and HamGNN [10]. For GaN, HamGNN data is absent due to out-of-memory error.

HfO2 with $4s2p2d2f1g$ basis, a typical model of 1.7M parameters can only predict the quantum tensors of structures up to $10^3$ atoms on devices with 32GB memory. Despite the linear dependency, the inference on a system with $10^4 \sim 10^7$ would require a memory with potentially over 300 GB, which is only available when parallelised to multiple GPUs. Therefore, a strictly localized model such as SLEM holds significant potential for expanding simulation system sizes.

## Discussion

This work proposed SLEM, the Strictly Local Equalvariant Message-Passing model for the prediction of quantum operator representations, as well as a systematic view of the parameterization of equivariant quantum operators. The strict local design of SLEM builds manybody dependence without expanding the effective receptive field, contributing to excellent accuracy, data efficiency and transferability. The integration of SO(2) convolution [19] enables rapid and memory-efficient tensor production, which is crucial for modeling heavy atoms. Furthermore, SLEM's locality potentially allows for extension to extremely large atomic systems through parallelization over partitioned atomic graphs, thus enabling first-principles electronic structure simulations at device-level scales. Beyond its modeling capabilities, SLEM intrinsically supports the representation of multiple quantum operators, including the Hamiltonian, density matrix, and overlap matrix. By parameterizing the overlap matrix using a two-center integration formula, it is transformed into invariant quantities readily obtainable from SLEM's initialized two-body embedding features. This

approach significantly reduces computational costs and minimizes the model's reliance on DFT software post-training.

While SLEM demonstrates efficacy across various applications, several areas warrant further investigation:

- Sampling Methodology for Active Learning: Scientific computation tasks require high confidence in calculated results, which can be challenging for data-driven approaches. Developing a robust sampling workflow for active learning is essential for reliability. While techniques like uncertainty-driven sampling with Gaussian regression [43] or model ensembles [44] have addressed confidence issues in machine learning force fields, designing a sampling workflow specifically for quantum operator models remains an open question.

- Software Integration for Post-Processing: Integrating the model with existing software is vital, especially for high-throughput calculations and large atomic systems beyond conventional DFT capabilities. An efficient, parallelizable solver for extracting physical quantities from the model's predictions is highly beneficial. While some software based on stochastic techniques [45, 46] shows promise, an open-sourced and highly-optimized solution for non-orthogonal bases remains unavailable.

Future research addressing these challenges will further enhance the applicability and reliability of SLEM and similar quantum operator models in computational materials science and chemistry.

## Code availability

The implementation of the SLEM model and its semi-local variant (named LEM) are open-source and accessible via the DeePTB GitHub repository. To facilitate the integration of DFT outputs with machine learning models, we have developed and released a supplementary tool, dftio. This tool enables parallel parsing of DFT outputs into a machine-readable data format, enhancing the efficiency of data preparation for model training and analysis.

## Acknowledgments

[1] P. Hohenberg and W. Kohn, Inhomogeneous electron gas, Phys. Rev. **136**, B864 (1964).

[2] W. Kohn and L. J. Sham, Self-consistent equations including exchange and correlation effects, Phys. Rev. **140**, A1133 (1965).

[3] R. O. Jones, Density functional theory: Its origins, rise to prominence, and future, Rev. Mod. Phys. **87**, 897 (2015).

[4] O. Unke, M. Bogojeski, M. Gastegger, M. Geiger, T. Smidt, and K.-R. Müller, SE(3)-equivariant prediction of molecular wavefunctions and electronic densities, Advances in Neural Information Processing Systems **34**, 14434 (2021).

[5] H. Yu, Z. Xu, X. Qian, X. Qian, and S. Ji, Efficient and equivariant graph networks for predicting quantum hamiltonian, in *International Conference on Machine Learning* (PMLR, 2023) pp. 40412–40424.

[6] X. Dong, E. Gull, and L. Wang, Equivariant neural network for green's functions of molecules and materials, Phys. Rev. B **109**, 075112 (2024).

[7] S. Yin, X. Pan, F. Wang, F. Wu, and L. He, A framework of so(3)-equivariant non-linear representation learning and its application to electronic-structure hamiltonian prediction, arXiv:2405.05722 (2024).

[8] X. Gong, H. Li, N. Zou, R. Xu, W. Duan, and Y. Xu, General framework for E(3)-equivariant neural network representation of density functional theory Hamiltonian, Nature Communications **14**, 2848 (2023).

[9] J. Nigam, M. J. Willatt, and M. Ceriotti, Equivariant representations for molecular Hamiltonians and N-center atomic-scale properties, The Journal of Chemical Physics **156**, 014115 (2022).

[10] Y. Zhong, H. Yu, M. Su, X. Gong, and H. Xiang, Transferable equivariant graph neural networks for the hamiltonians of molecules and solids, npj Computational Materials **9**, 182 (2023).

[11] H. Li, Z. Wang, N. Zou, M. Ye, R. Xu, X. Gong, W. Duan, and Y. Xu, Deep-learning density functional theory hamiltonian for efficient ab initio electronic-structure calculation, Nature Computational Science **2**, 367 (2022).

[12] J. Han, J. Cen, L. Wu, Z. Li, X. Kong, R. Jiao, Z. Yu, T. Xu, F. Wu, Z. Wang, *et al.*, A survey of geometric graph neural networks: Data structures, models and applications, arXiv:2403.00485 (2024).

[13] A. Musaelian, S. Batzner, A. Johansson, L. Sun, C. J. Owen, M. Kornbluth, and B. Kozinsky, Learning local equivariant representations for large-scale atomistic dynamics, Nature Communications **14**, 579 (2023).

[14] I. Batatia, D. P. Kovacs, G. Simm, C. Ortner, and G. Csányi, MACE: Higher order equivariant message passing neural networks for fast and accurate force fields, Advances in Neural Information Processing Systems **35**, 11423 (2022).

[15] C. K. Joshi, C. Bodnar, S. V. Mathis, T. Cohen, and P. Lio, On the expressive power of geometric graph neural networks, in *International conference on machine learning* (PMLR, 2023) pp. 15330–15355.

[16] Y.-L. Liao and T. Smidt, Equiformer: Equivariant graph attention transformer for 3d atomistic graphs, arXiv:2206.11990 (2022).

[17] Y.-L. Liao, B. Wood, A. Das, and T. Smidt, Equiformerv2: Improved equivariant transformer for scaling to higher-degree representations, arXiv:2306.12059 (2023).

[18] G. Simeon and G. De Fabritiis, TensorNet: Cartesian tensor representations for efficient learning of molecular potentials, Advances in Neural Information Processing Systems **36** (2024).

[19] S. Passaro and C. L. Zitnick, Reducing SO(3) convolutions to SO(2) for efficient equivariant gnns, in *International Conference on Machine Learning* (PMLR, 2023) pp. 27420–27438.

[20] L. Zitnick, A. Das, A. Kolluru, J. Lan, M. Shuaibi, A. Sriram, Z. Ulissi, and B. Wood, Spherical channels for modeling atomic interactions, Advances in Neural Information Processing Systems **35**, 8054 (2022).

[21] E. Huckel and P. Debye, Zur theorie der elektrolyte. i. gefrierpunktserniedrigung und verwandte erscheinungen, Phys. Z **24**, 185 (1923).

[22] R. Resta, Thomas-fermi dielectric screening in semiconductors, Phys. Rev. B **16**, 2717 (1977).

[23] D. Ninno, F. Trani, G. Cantele, K. J. Hameeuw, G. Iadonisi, E. Degoli, and S. Ossicini, Thomas-Fermi model of electronic screening in semiconductor nanocrystals, Europhysics Letters **74**, 519 (2006).

[24] Q. Gu, Z. Zhouyin, S. K. Pandey, P. Zhang, L. Zhang, and W. E, DeePTB: A deep learning-based tight-binding approach with *ab initio* accuracy, arXiv:2307.04638 (2023).

[25] J. C. Slater and G. F. Koster, Simplified LCAO method for the periodic potential problem, Phys. Rev. **94**, 1498 (1954).

[26] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, Neural message passing for quantum chemistry, in *International conference on machine learning* (PMLR, 2017) pp. 1263–1272.

[27] K. T. Schütt, S. Chmiela, O. A. Von Lilienfeld, A. Tkatchenko, K. Tsuda, and K.-R. Müller, Machine learning meets quantum physics, Lecture Notes in Physics (2020).

[28] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, SchNet – A deep learning architecture for molecules and materials, The Journal of Chemical Physics **148**, 241722 (2018).

[29] V. G. Satorras, E. Hoogeboom, and M. Welling, E(n) equivariant graph neural networks, in *International conference on machine learning* (PMLR, 2021) pp. 9323–9332.

[30] J. Behler and M. Parrinello, Generalized neural-network representation of high-dimensional potential-energy surfaces, Phys. Rev. Lett. **98**, 146401 (2007).

[31] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons, Phys. Rev. Lett. **104**, 136403 (2010).

[32] A. Thompson, L. Swiler, C. Trott, S. Foiles, and G. Tucker, Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials, Journal of Computational Physics **285**, 316 (2015).

[33] L. Zhang, J. Han, H. Wang, R. Car, and W. E, Deep potential molecular dynamics: A scalable model with the accuracy of quantum mechanics, Phys. Rev. Lett. **120**, 143001 (2018).

[34] H. Wang, L. Zhang, J. Han, and W. E, DeePMD-kit: A deep learning package for many-body potential energy representation and molecular dynamics, Computer Physics Communications **228**, 178 (2018).

[35] L. Zhang, J. Han, H. Wang, W. Saidi, R. Car, and W. E, End-to-end symmetry preserving inter-atomic potential energy model for finite and extended systems, Advances in Neural Information Processing Systems **31** (2018).

[36] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley, Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds, arXiv:1802.08219 (2018).

[37] M. Weiler, M. Geiger, M. Welling, W. Boomsma, and T. S. Cohen, 3D Steerable CNNs: Learning rotationally equivariant features in volumetric data, Advances in Neural Information Processing Systems **31** (2018).

[38] R. Kondor, Z. Lin, and S. Trivedi, Clebsch–Gordan nets: a fully fourier space spherical convolutional neural network, Advances in Neural Information Processing Systems **31** (2018).

[39] R. Kondor, N-body networks: a covariant hierarchical neural network architecture for learning atomic potentials, arXiv:1803.01588 (2018).

[40] A. V. Podolskiy and P. Vogl, Compact expression for the angular dependence of tight-binding Hamiltonian matrix elements, Phys. Rev. B **69**, 233101 (2004).

[41] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials, Nature Communications **13**, 2453 (2022).

[42] M. Geiger and T. Smidt, e3nn: Euclidean neural networks, arXiv:2207.09453 (2022).

[43] J. Vandermause, S. B. Torrisi, S. Batzner, Y. Xie, L. Sun, A. M. Kolpak, and B. Kozinsky, On-the-fly active learning of interpretable Bayesian force fields for atomistic rare events, npj Computational Materials **6**, 1 (2020).

[44] Y. Zhang, H. Wang, W. Chen, J. Zeng, L. Zhang, H. Wang, and W. E, DP-GEN: A concurrent learning platform for the generation of reliable deep learning based potential energy models, Computer Physics Communications **253**, 107206 (2020).

[45] Y. Li, Z. Zhan, X. Kuang, Y. Li, and S. Yuan, TBPLaS: A tight-binding package for large-scale simulation, Computer Physics Communications **285**, 108632 (2023).

[46] S. M. João, M. Anđelković, L. Covaci, T. G. Rappoport, J. M. Lopes, and A. Ferreira, Kite: high-performance accurate modelling of electronic structure and response functions of large molecules, disordered crystals and heterostructures, Royal Society Open Science **7**, 191809 (2020).