

Efficient Prediction of SO(3)-Equivariant Hamiltonian Matrices via SO(2) Local Frames

Haiyang Yu¹ Yuchao Lin¹ Xuan Zhang¹ Xiaofeng Qian^{2, 3, 4*} Shuiwang Ji^{1*†}

¹Department of Computer Science and Engineering, Texas A&M University

²Department of Materials Science and Engineering, Texas A&M University

³Department of Electrical and Computer Engineering, Texas A&M University

⁴Department of Physics and Astronomy, Texas A&M University

Abstract

We consider the task of predicting Hamiltonian matrices to accelerate electronic structure calculations, which plays an important role in physics, chemistry, and materials science. Motivated by the inherent relationship between the off-diagonal blocks of the Hamiltonian matrix and the SO(2) local frame, we propose a novel and efficient network, called QHNetV2, that achieves global SO(3) equivariance without the costly SO(3) Clebsch–Gordan tensor products. This is achieved by introducing a set of new efficient and powerful SO(2)-equivariant operations and performing all off-diagonal feature updates and message passing within SO(2) local frames, thereby eliminating the need of SO(3) tensor products. Moreover, a continuous SO(2) tensor product is performed within the SO(2) local frame at each node to fuse node features, mimicking the symmetric contraction operation. Extensive experiments on the large QH9 and MD17 datasets demonstrate that our model achieves superior performance across a wide range of molecular structures and trajectories, highlighting its strong generalization capability. The proposed SO(2) operations on SO(2) local frames offer a promising direction for scalable and symmetry-aware learning of electronic structures. Our code will be released as part of the AIRS library (<https://github.com/divelab/AIRS>).

1 Introduction

Quantum Hamiltonian, as a central element in the many-body Schrödinger equation of quantum mechanics, plays a key role in governing the quantum states and physical properties of molecules and materials, making it essential for physics, chemistry, and materials science. First-principles methods such as density functional theory (DFT) [1, 2, 3] have been developed to solve the Schrödinger equation and investigate the electronic structures of molecules and solids. Despite their success, these methods struggle with high computational cost, limiting their application to systems with only a few hundred atoms. In DFT, the central Kohn-Sham equation is solved using the self-consistent field (SCF) method based on the variational principle of the second Hohenberg-Kohn theorem [2]. The electronic wavefunctions, energies, charge density, and Kohn-Sham Hamiltonian are iteratively calculated and updated until convergence is achieved. This SCF calculation has a high time complexity of $O(N_e^3T)$, where N_e is the number of electrons and T is the number of SCF steps required for convergence. Consequently, DFT remains computationally expensive for large and diverse quantum systems. More background details about DFT can be found in Appendix A.1.

*Equal senior contribution

†Correspondence to: Xiaofeng Qian <feng@tamu.edu> and Shuiwang Ji <sji@tamu.edu>

Recently, deep learning has demonstrated great potential in advancing scientific research [4, 5, 6, 7]. In particular, several machine learning models [8, 9, 10, 11, 12, 13, 14, 15] have been developed to directly predict the Hamiltonian matrix from atomic structures, achieving substantial speedup of several orders of magnitude in inference time compared to traditional DFT calculations. Despite these advances, achieving higher accuracy and more efficient training remains a key challenge in the development of quantum tensor networks for predicting Hamiltonian matrices and many other multi-physical coupling matrices.

To address the above challenge, here we propose a novel and efficient network QHNetV2, motivated by the inherent relationship between the off-diagonal blocks of the Hamiltonian matrix and the SO(2) local frame to achieve global SO(3) equivariance without using the computationally intensive SO(3) Clebsch–Gordan tensor products. Specifically, we first eliminate the need of SO(3) tensor products by introducing a set of new efficient and powerful SO(2)-equivariant operations and performing all off-diagonal feature updates and message passing within SO(2) local frames. Second, to effectively perform nonlinear node updates, we apply a continuous SO(2) tensor product within the SO(2) local frame at each node to fuse node features, mimicking the symmetric contraction module used in MACE [16] for modeling many-body interactions. We conduct extensive experiments on the large QH9 and MD17 datasets, which shows the superior performance and strong generalization capability of our new framework over a diverse set of molecular structures and trajectories. Additionally, these novel SO(2) operations on SO(2) local frames presents a promising avenue for scalable and symmetry-aware learning of electronic structures.

2 Preliminaries

2.1 Hamiltonian Matrices and SO(3) Equivariance

The Kohn-Sham equation of molecular and materials systems is given by: $\hat{H}_{\text{KS}}|\psi_n\rangle = \epsilon_n|\psi_n\rangle$, where \hat{H}_{KS} is the Kohn-Sham Hamiltonian operator, ψ_n is single-electron eigen wavefunction (also called molecular orbital), and ϵ_n is the corresponding eigen energy. Under a predefined basis set such as the STO-3G atomic orbital basis $\{\phi_p\}$ combining radial functions with spherical harmonics, the Kohn-Sham equation can be converted into a matrix form: $\mathbf{H}\mathbf{C} = \mathbf{\epsilon}\mathbf{S}\mathbf{C}$, with Hamiltonian matrix element $\mathbf{H}_{pq} = \int \phi_p^*(\mathbf{r})\hat{H}_{\text{KS}}\phi_q(\mathbf{r})d\mathbf{r}$ and overlap matrix element $\mathbf{S}_{pq} = \int \phi_p^*(\mathbf{r})\phi_q(\mathbf{r})d\mathbf{r}$. $\mathbf{\epsilon}$ is a diagonal matrix of the eigen energies, and \mathbf{C} contains wavefunction coefficients, with each eigen wavefunction ψ_n as a linear combination of the basis functions $\psi_n(\mathbf{r}) = \sum_p \mathbf{C}_{pn}\phi_p(\mathbf{r})$ [17]. Hamiltonian matrix follows block-wise SO(3) equivariance upon 3D rotations. More details can be found in Appendices A.1, A.2, and A.3.

When learning the Hamiltonian matrix using machine learning models, it is essential to incorporate high-degree equivariant features that align with the angular momentum quantum numbers ℓ of atomic orbitals. This is particularly important for accurately capturing orbitals with high angular momentum such as d -orbitals with $\ell = 2$, requiring the highest degree of SO(3) equivariant features with $L_{\text{max}} \geq 4$. Furthermore, the pairwise interactions are needed for predicting all Hamiltonian matrix blocks while maintaining the block-wise SO(3) equivariance.

Machine learning demonstrates its power for property predictions, force field development, and so on [18, 19, 20, 21, 22, 23, 24]. And numerous models have been built upon tensor product (TP) have proven to be an effective approach [25, 26, 23, 27, 28, 29, 27, 16, 30, 31, 32, 33] with high-degree equivariant features. However, the computational complexity of TPs is $O(L_{\text{max}}^6)$ which increases significantly with the maximum degree L_{max} , posing a substantial challenge for tasks such as Hamiltonian matrix prediction where a high L_{max} is often required. A promising alternative is to replace full TPs by SO(2) convolutions, as proposed in eSCN [34]. This approach demonstrates that the SO(2) Linear operation can be equivalent to SO(3) TP while reducing computational complexity to $O(L_{\text{max}}^3)$. This raises an intriguing question: can more and diverse SO(2)-based operations be used to model the Hamiltonian matrix more effectively and accurately, especially in cases where a high L_{max} is required?

2.2 SO(3) and SO(2) Irreducible Representations

Under an arbitrary 3D rotation, the atomic orbital basis rotates accordingly. As a result, the Hamiltonian matrix element defined on the atomic orbital pairs transforms in an equivariant manner. It

is therefore necessary to introduce irreducible representations (irreps) of the 3D rotation group, *i.e.* SO(3) group. The SO(3) group consists of all 3D rotations, represented by 3×3 orthogonal matrices with determinant 1. For each SO(3) irrep with degree $\ell \in \mathbb{N}_0$, the corresponding representation space has dimension $2\ell + 1$, with spherical harmonics $\{Y_m^\ell(\theta, \phi)\}_{m=-\ell}^\ell$ serving as a natural basis. For each group element $g \in \text{SO}(3)$ (*i.e.* 3D rotation), the corresponding matrix representation for the ℓ -th irrep is the Wigner D-matrix $D^\ell(g)$ of shape $(2\ell + 1) \times (2\ell + 1)$. Then, $D^\ell(g)$ is applied via matrix multiplication to perform the corresponding SO(3) rotation in the representation space. To develop more advanced operations within the SO(2) symmetry space, it is essential to introduce irreducible representations (irreps) of the SO(2) group. The SO(2) group describes 2D planar rotations, which can be interpreted as rotations around a fixed axis in 3D space (typically the z-axis). The irreps of SO(2) has dimension 2 for $m > 0$ and 1 for $m = 0$ or can be represented as complex values, with order $m \in \mathbb{N}_0$. A natural basis function for SO(2) irreps is the set of real circular harmonics, which take the form $B^m(\delta) = [\sin(m\delta), \cos(m\delta)]^T$ for $m \geq 1$ and $B^0(\delta) = [1]$ for $m = 0$. Each group element $g \in \text{SO}(2)$ corresponds to a rotation by an angle $\varphi \in [0, 2\pi)$, and the matrix representation for the m -th irrep is given by $\begin{pmatrix} \cos(m\varphi) & \sin(m\varphi) \\ -\sin(m\varphi) & \cos(m\varphi) \end{pmatrix}$ for $m > 0$, and 1 for $m = 0$. Subsequently, the representation space under SO(2) rotation transforms via matrix multiplication. In the complex circular harmonics basis $e^{im\delta}$, the representation space under SO(2) rotation transforms via multiplication with $e^{im\varphi}$.

3 Methods

This section introduces the model we have built for the Hamiltonian matrix prediction task. The key is to develop the SO(2) local frame, where any powerful SO(2) equivariant operations can be applied while maintaining the overall framework to be SO(3) equivariant. Later on, we demonstrate that the model based on SO(2) local frame achieve great efficiency and accuracy, especially these two key things for building this networks especially considering the high L_{\max} is an unavoidable things.

3.1 SO(2) Local Frames

In this section, we introduce the concept of the SO(2) local frame, extending the formulation from the operation in eSCN [34], and further demonstrate that the overall network preserves SO(3) equivariance if the computations within the SO(2) local frame are SO(2) equivariant. This holds not only for SO(2) Linear layers but also for more general SO(2)-equivariant operations, achieved through frame averaging.

The overall framework consists of a global coordinate system and local SO(2) frames. Within the global coordinate system, the model operates on SO(3) irreps and maintains SO(3) equivariance. Therefore, the input to each local network Φ is SO(3) equivariant. We then apply the minimal frame averaging [35] on the input irreps to maintain equivariance of Φ . More details for frame averaging [36, 37, 35] can be found in Appendix B. To perform minimal frame averaging, canonicalization [35] on the input irreps is required before feeding into Φ . Here, we define the canonicalization operation based on a local unit vector \hat{r} as $c(\hat{r}) = \hat{v}$, where a rotation $h \in \text{SO}(3)$ is applied to align the local unit vector \hat{r} with the canonical vector \hat{v} . Furthermore, the local frame can be defined as $\mathcal{F}(\hat{r})$ which takes \hat{r} as the local vector mapping from \mathbb{R}^3 and the irreps built upon this to SO(3) space with SO(2)-equivariant local model Φ . And the corresponding rotation h is defined as $\mathcal{FR}(\hat{r}, \cdot)$ while the inverse rotation h^{-1} is defined as $\mathcal{FR}^{-1}(\hat{r}, \cdot)$. Inspired by the eSCN [34], we choose the canonical vector $\hat{v} = [0, 1, 0]$ and local frame $\mathcal{F}(\hat{r}_{ij})$. In this case, when Φ is a SO(2) Linear, it becomes equivalent to SO(3) tensor product between the input irreps and spherical harmonics of pairwise direction $Y(\hat{r}_{ij})$. Note that the message passing mechanism of eSCN is a specific instance covered by our proposed framework. And to our knowledge, we are the first to link the eSCN operation to frame averaging.

3.2 SO(2) Equivariant Operations

Based on the above analysis of SO(2) local frame, any SO(2)-equivariant layer can be applied within this frame while preserving the overall SO(3)-equivariance of the architecture. In this subsection, we discuss several SO(2) equivariant building blocks used within the SO(2) local frame.

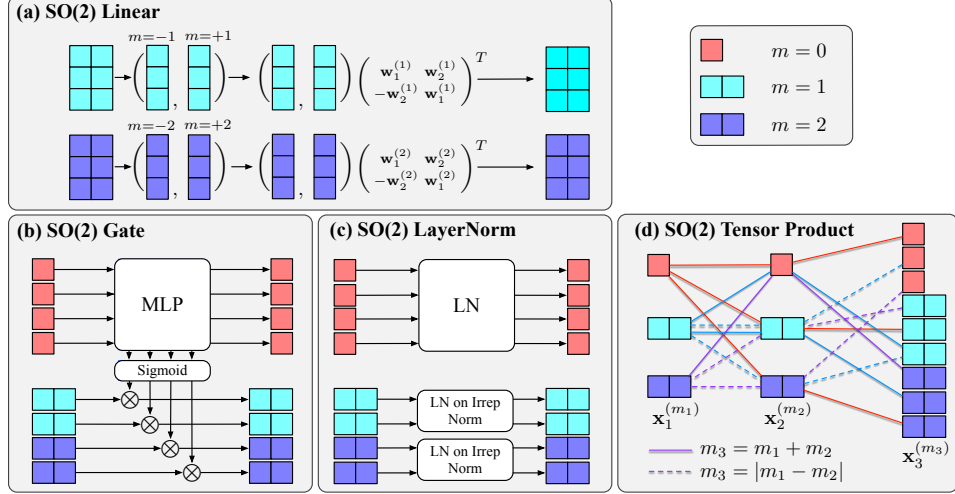


Figure 1: SO(2) equivariant operations. (a) SO(2) Linear. For SO(2) irreps with order $m > 0$, this operation uses weight matrices $\mathbf{w}_1^{(m)}, \mathbf{w}_2^{(m)} \in \mathbb{R}^{C \times C}$ where C is the number of channels for input irreps. (b) SO(2) Gate. For the $m = 0$ features, a multi-layer perceptrons (MLP) is used to update them. Simultaneously, for each irrep with order $m > 0$, the MLP outputs a gate value passed through a sigmoid function, which modulates the corresponding SO(2)-equivariant features. (c) SO(2) LayerNorm (LN). For the $m = 0$ features, a standard LN is applied. For $m > 0$ features, LN is applied on the norm of SO(2) irreps according to Eq. 4. (d) SO(2) Tensor Product (TP). The SO(2) tensor product fuses features by combining irreps under the constraints $m_3 = m_1 + m_2$ (shown as solid lines) or $m_3 = |m_1 - m_2|$ (shown as dashed lines). The color of the path corresponds to its originating SO(2) irreps. A more general case containing $v - 1$ TPs for v set of SO(2)-equivariant features is shown in Eq. 8. Each valid combination defines a path, ensuring the resulting features remain SO(2)-equivariant.

SO(2) Linear. SO(2) Linear layer is first introduced in eSCN [34], and used in following works like EquiformerV2 [38]. This linear operation takes SO(2) irreducible representations \mathbf{x} as input, and applies the multiplication between the \mathbf{x} and weights \mathbf{w} with the formulation defined as

$$\begin{pmatrix} \mathbf{z}_{c,-m} \\ \mathbf{z}_{c,m} \end{pmatrix} = \sum_{c'} \begin{pmatrix} \mathbf{w}_{1,cc'}^{(m)} & \mathbf{w}_{2,cc'}^{(m)} \\ -\mathbf{w}_{2,cc'}^{(m)} & \mathbf{w}_{1,cc'}^{(m)} \end{pmatrix} \begin{pmatrix} \mathbf{x}_{c',-m} \\ \mathbf{x}_{c',m} \end{pmatrix}, \quad (1)$$

where c' is the channel index for input \mathbf{x} and c is the channel index for output features \mathbf{z} . This linear operation can be easily understood if we convert them into complex numbers. Specifically, we use $\tilde{\mathbf{x}}_{c'}^{(m)} = \mathbf{x}_{c',m} + i\mathbf{x}_{c',-m} = \bar{\mathbf{x}}_{c'}^{(m)} e^{i\theta_{x(m),c'}}$ and weights $\tilde{\mathbf{w}}_{cc'}^{(m)} = \mathbf{w}_{1,cc'}^{(m)} + i\mathbf{w}_{2,cc'}^{(m)} = \bar{\mathbf{w}}_{cc'}^{(m)} e^{i\theta_{w(m),cc'}}$. The above equation is equivalent to

$$\tilde{\mathbf{z}}_c^{(m)} = \sum_{c'} \tilde{\mathbf{w}}_{cc'}^{(m)} \tilde{\mathbf{x}}_{c'}^{(m)} = \sum_{c'} \bar{\mathbf{x}}_{c'}^{(m)} \bar{\mathbf{w}}_{cc'}^{(m)} e^{i(\theta_{w(m),cc'} + \theta_{x(m),c'})}, \quad (2)$$

where $\tilde{\mathbf{z}}_c^{(m)} = \mathbf{z}_{c,m} + i\mathbf{z}_{c,-m}$. It denotes a complex linear layer without bias term, which includes internal weights on the scale $\bar{\mathbf{w}}^{(m)}$ and rotation with angle $\theta_{w(m)}$, as well as self-interaction across all input channels. Note that there is a set of weights for each individual \mathbf{x}^m with $m > 0$.

SO(2) Gate. Gate activation is a useful component on various networks for SO(3) irreducible representations [28, 38], and this operation can also be applied to the SO(2) irreducible representations. Specifically, the corresponding formulation is shown as

$$\mathbf{z}^{(m)} = \begin{cases} \text{MLP}(\mathbf{x}^{m=0}) & , \text{ if } m = 0, \\ \text{Sigmoid}(\text{MLP}(\mathbf{x}^{m=0})) \circ \mathbf{x}^{(m)} & , \text{ if } m > 0, \end{cases} \quad (3)$$

An MLP is applied to learn the $m = 0$ features, and the other MLP takes the $m = 0$ features as the input, applies a sigmoid gate activation, then multiplies with the $m > 0$ features to control the irreducible features.

SO(2) Layer Normalization. Within the overall framework, it is often necessary to maintain a set of SO(2)-equivariant features to model the target quantities. In our task, for each off-diagonal block of the Hamiltonian matrix, it is required to model its corresponding SO(2) features. Since LayerNorm [39] is an important technique to stabilize the training procedure, we extend to apply it to SO(2) irreducible representations $\mathbf{x}^{(m)}$, defined as

$$\text{LN}(\mathbf{x}^{(m)}) = \frac{\mathbf{x}^{(m)}}{\text{norm}(\mathbf{x}^{(m)})} \circ \left(\frac{\text{norm}(\mathbf{x}^{(m)}) - \mu^{(m)}}{\sigma^{(m)}} \circ g^{(m)} + b^{(m)} \right), \quad (4)$$

where $\mu^{(m)} = \frac{1}{C} \sum_{i=0}^C \text{norm}(\mathbf{x}_i^{(m)})$, $\mu^{(m)} \in \mathbb{R}^{N \times 1 \times 1}$ is the mean of the norm across the channels and $\sigma^{(m)} = \sqrt{\frac{1}{C} \sum_{i=0}^C (\text{norm}(\mathbf{x}_i^{(m)}) - \mu^{(m)})^2}$, $\sigma^{(m)} \in \mathbb{R}^{N \times 1 \times 1}$ is the standard derivation of norm over the channels. After normalization, learnable affine parameters are introduced with a scale factor $g^{(m)} \in \mathbb{R}^{1 \times 1 \times C}$ and a bias term $b^{(m)} \in \mathbb{R}^{1 \times 1 \times C}$, which are used to rescale and recenter the scale of the SO(2) equivariant features with $m > 0$. We refer to it as norm-based Layer Normalization (LN), consistent with Eq. 4.

SO(2) Tensor Product (TP). Similar to the tensor product used for SO(3) features, the SO(2) tensor product is also a valid operation for fusing SO(2) features and is closely related to performing SO(2) convolution on the circle. Specifically, the formulation is given by

$$\mathbf{z}^{(m_o)} = \mathbf{x}_1^{(m_1)} \otimes \mathbf{x}_2^{(m_2)} = \begin{pmatrix} \mathbf{x}_{1,-m_1}^{(m_1)} \mathbf{x}_{2,-m_2}^{(m_2)} \\ \mathbf{x}_{1,m_1}^{(m_1)} \mathbf{x}_{2,m_2}^{(m_2)} \end{pmatrix}, \quad (5)$$

where $m_o = m_1 + m_2$. Similarly, when $m_o = m_1 - m_2$ with $m_1 > m_2$, the corresponding SO(2) TP can be formulated as

$$\mathbf{z}^{(m_o)} = \begin{pmatrix} \mathbf{x}_{1,-m_1}^{(m_1)} \mathbf{x}_{2,m_2}^{(m_2)} \\ \mathbf{x}_{1,m_1}^{(m_1)} \mathbf{x}_{2,-m_2}^{(m_2)} \end{pmatrix} = \text{flip}(\mathbf{x}_1^{(m_1)}, +1) \otimes \text{flip}(\mathbf{x}_2^{(m_2)}, -1), \quad (6)$$

where

$$\text{flip}\left(\begin{pmatrix} \mathbf{x}_m^{(m)} \\ \mathbf{x}_{-m}^{(m)} \end{pmatrix}, +1\right) = \begin{pmatrix} \mathbf{x}_m^{(m)} \\ \mathbf{x}_{-m}^{(m)} \end{pmatrix} \text{ and } \text{flip}\left(\begin{pmatrix} \mathbf{x}_m^{(m)} \\ \mathbf{x}_{-m}^{(m)} \end{pmatrix}, -1\right) = \begin{pmatrix} \mathbf{x}_{-m}^{(m)} \\ \mathbf{x}_m^{(m)} \end{pmatrix}, \quad (7)$$

denoting whether to flip the index of order m for the SO(2) irreducible representation. Then, for an SO(2) TP operation on a set of SO(2) irreducible representations, it should include all valid interaction paths, those satisfying $m_o = m_1 + m_2$, $m_o = m_1 - m_2$, $m_1 > m_2$, and $m_o = m_2 - m_1$, $m_2 > m_1$. Given the maximum order m_{max} , the number of paths is $O(m_{max}^2)$.

Moreover, inspired by the symmetric contraction module proposed in MACE [16] which uses the generalized CG to fuse multiple SO(3)-equivariant features after aggregating the node features to perform many-body interactions, a continuous SO(2) tensor product can be implemented in a similar fashion to capture more complex interactions. Specifically, we consider the interactions among v sets of SO(2)-equivariant features and for each path with constraint $m_o = s_1 m_1 + s_2 m_2 + \dots + s_v m_v$ where s is the sign value within the range $\{-1, +1\}$, the corresponding output of this path is

$$\mathbf{z}^{m_o} = \underbrace{\text{flip}(\mathbf{x}^{(m_1)}, s_1) \otimes \text{flip}(\mathbf{x}^{(m_2)}, s_2) \otimes \dots \otimes \text{flip}(\mathbf{x}^{(m_v)}, s_v)}_{(v-1) \text{ TP}}. \quad (8)$$

Considering a set of SO(2) irreducible representations with maximum order m_{max} , the number of paths for SO(2) TP is $O(m_{max}^v)$.

3.3 Model Architecture

With the above SO(2) equivariant operations, here we introduce our model based on these SO(2) local frames and the global coordinate system for the Hamiltonian matrix prediction task.

Node Embedding and Node Pair Embedding. The node embedding learns one embedding for each atomic type. The node pair embedding takes the node embeddings for both nodes as well as the pairwise distance, denoted as

$$\mathbf{x}_{ij}^{m=0} = \text{MLP}(\text{Linear}(\mathbf{s}_{ij}) \circ \text{Linear}(\text{rbf}(\bar{r}_{ij}))) \quad (9)$$

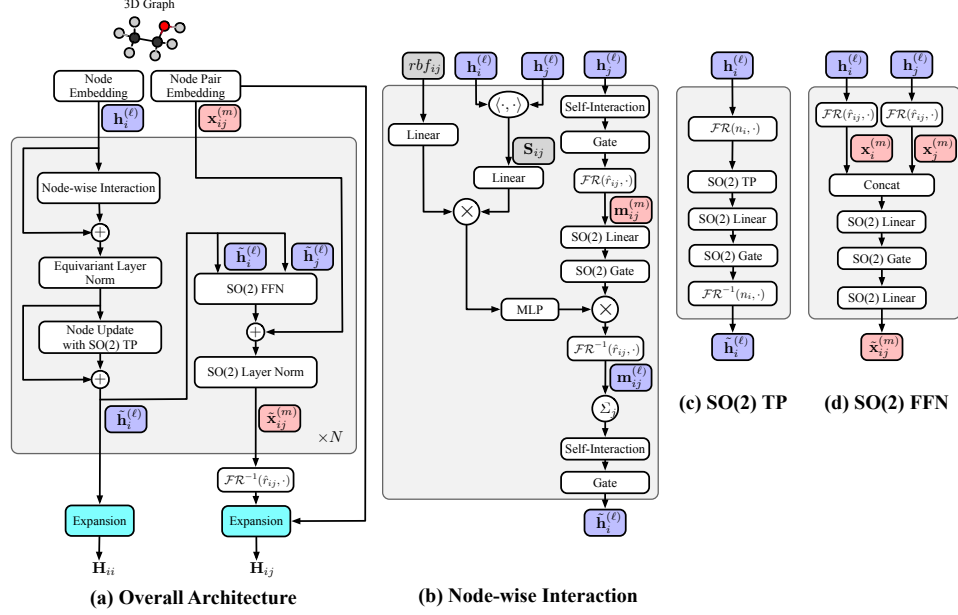


Figure 2: The overall architecture of the proposed QHNetV2. In this figure, \times denotes element-wise multiplication, $\langle \cdot, \cdot \rangle$ denotes inner product. Gray color denotes scalar values, red color denotes SO(2) irreps, and blue color denotes SO(3) irreps.

Node-wise interaction. For the node-wise interaction module, the message passing paradigm of Graph Neural Networks (GNNs) [40] is applied to aggregate node features from their neighbors while preserving equivariance. As illustrated in Figure 2(b), a self-interaction layer and gate [25] operation is first applied to the input SO(3) irreps $\mathbf{h}_j^{(\ell)}$. After that, for each message, the $\mathbf{h}_j^{(\ell)}$ is rotated into its local SO(2) frame $\mathcal{F}(r_{ij})$ obtaining the SO(2) irreps $\mathbf{m}_{ij}^{(m)}$. Then, SO(2) Linear and SO(2) Gate operations are applied to the message $\mathbf{m}_{ij}^{(m)}$. The radial basis function (rbf), encoding the pairwise distance information, denoted as rbf_{ij} , is combined with the inner product of the equivariant features from both source and target nodes to multiply with $\tilde{\mathbf{m}}_{ij}$. Specifically, the inner product of node features is defined as

$$\mathbf{s}_{ij} = \left(\langle \mathbf{h}_i^{\ell=0}, \mathbf{h}_j^{\ell=0} \rangle \| \langle \mathbf{h}_i^{\ell=1}, \mathbf{h}_j^{\ell=1} \rangle \| \cdots \| \langle \mathbf{h}_i^{\ell=\ell_{max}}, \mathbf{h}_j^{\ell=\ell_{max}} \rangle \right), \quad (10)$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product and $\|$ represents vector concatenation across different SO(3) degrees. Then, the message is defined as

$$\tilde{\mathbf{m}}_{ij}^{(m)} = \text{SO}(2)\text{Linear}(\mathbf{m}_{ij}^{(m)}) \circ \text{MLP}(\text{Linear}(\mathbf{s}_{ij}) \circ \text{Linear}(\text{rbf}(\bar{r}_{ij}))). \quad (11)$$

Subsequently, $\tilde{\mathbf{m}}_{ij}^{(m)}$ is rotated back to the global coordinate system to get message $\mathbf{m}_{ij}^{(\ell)}$ which is then aggregated together. A self-interaction with gate is applied on the aggregated node irreps.

Node feature updating with SO(2) Tensor Product (TP). Motivated by the symmetric contraction module proposed in MACE, a node updating module is applied with consecutive SO(2) TPs after the aggregations, as shown in Figure 2(c).

First, we need to define the local frame for each node. Specifically, for node n_i , we select the direction vector from the closest neighbor node to the center node to build the local SO(2) frame, denoted as

$$\mathcal{F}(n_i) = \mathcal{F}(\hat{r}_{ij}) \text{ with } \arg \min_{j \in \mathcal{N}_i} \bar{r}_{ij}, \quad (12)$$

where \bar{r}_{ij} is the pairwise distance. Then, the node feature updating module is performed as

$$\mathbf{x}_i^{(m)} = \mathcal{FR}(n_i, \mathbf{h}_i^{(\ell)}), \tilde{\mathbf{x}}_i^{(m)} = \text{SO}(2)\text{Linear}(\text{SO}(2)\text{TP}(\mathbf{x}_i^{(m)}, v)), \tilde{\mathbf{h}}_i^{(\ell)} = \mathcal{FR}^{-1}(n_i, \tilde{\mathbf{x}}_i^{(m)}), \quad (13)$$

where the SO(2) TP operation collects the SO(2) irreps according to Eq. 8 with $v - 1$ consecutive SO(2) TP operations. The collected SO(2) irreps are then fed into a SO(2) Linear before being transformed back to the global coordinate system.

Off-diagonal feature updating with SO(2) Feed Forward Networks (FFNs). The off-diagonal features account for the majority of the final Hamiltonian matrix, and there is a one-one correspondence between the SO(2) local frame and its off-diagonal matrix block. Therefore, we keep all the features within this SO(2) local frame over the layers.

For each layer shown in Figure 2(d), the features coming from the neighbors $\mathbf{h}_i^{(\ell)}$ and $\mathbf{h}_j^{(\ell)}$ are transferred to the local frame $\mathcal{F}(\mathbf{r}_{ij})$, obtaining the corresponding SO(2) irreps $\mathbf{m}_i^{(m)}$ and $\mathbf{m}_j^{(m)}$. Hence, the associated off-diagonal feature is given by

$$\mathbf{x}_{ij}^{(m)} = \text{SO(2) Linear} \left(\text{SO(2) Gate} \left(\text{SO(2) Linear}(\mathbf{m}_i^{(m)} \parallel \mathbf{m}_j^{(m)}) \right) \right) \quad (14)$$

Overall architecture. The overall architecture is demonstrated in Figure 2(a). It contains two parts. The left one updates the node features $\mathbf{h}_i^{(\ell)}$. The skip connection is applied after the node-wise interaction, and then, the Equivariant LayerNorm [38] is applied before feeding into the SO(2) TP operations. The right one is used to update the pairwise features $\mathbf{x}_{ij}^{(m)}$. With the updated pairwise node features $\tilde{\mathbf{h}}_i$ and $\tilde{\mathbf{h}}_j$, it applies the SO(2) FFN on them, followed by a skip connection with SO(2) LayerNorm to update the pairwise SO(2) irreps $\tilde{\mathbf{x}}_{ij}^{(m)}$.

Matrix Construction. We follow the expansion module in QHNet [10], which uses the $(\bar{\otimes} w_3)^{(\ell_1, \ell_2)}_{(m_1, m_2)} = \sum_{m_3=-\ell_3}^{\ell_3} C_{(\ell_3, m_3)}^{(\ell_1, m_1), (\ell_2, m_2)} w_{m_3}^{\ell_3}$ and establishes a mapping between the irrep blocks and the orbital pairs within the Hamiltonian matrix blocks. For the diagonal matrix block, the node feature $\tilde{\mathbf{h}}^{(\ell)}$ is directly used as the input of Expansion module. For the off-diagonal matrix blocks, the SO(2) irreps are first transformed back to the global coordinate system and then fed into the Expansion module.

4 Related Work

Multiple studies have explored the task of Hamiltonian matrix prediction in quantum systems. SchNorb [41] takes the invariant SchNet [18] as its backbone and build models for directly predicting the Hamiltonian matrix blocks. Subsequent models such as PhiSNet [8] and QHNet [10] incorporate equivariance into Hamiltonian modeling by leveraging tensor product (TP) operations and equivariant neural networks to respect the underlying rotational symmetries of the system. SPHNet [14] introduces sparse gate on tensor product mechanism to accelerate equivariant computations while maintaining competitive performance. WANet [13] addresses the nonlinearity of the Hamiltonian’s eigenvalues—*i.e.*, the energy spectrum—by introducing an auxiliary loss function that directly constrains the predicted eigenvalues, enhancing both accuracy and physical consistency. For materials datasets, the DeepH series of works [42, 43, 44, 45, 46] framework demonstrates the evolution from invariant to fully equivariant architectures, capable of predicting both Hamiltonians and higher-order tensors along material trajectories. HamGNN [12] takes use of TP to build the equivariant graph networks for its prediction, and DeePTB [15] designs a strictly local model that efficiently captures many-body interactions. As a plug-in module, TraceGrad [47] introduces an auxiliary objective based on the norm of the predicted Hamiltonian blocks and corresponding blocks for improving the performance of Hamiltonian matrix prediction. For the self-consistent training framework [48, 49, 50], it combines the machine models with a self-consistent field (SCF) loop during training, enabling unsupervised learning of quantum properties. Although this method enhances accuracy and expressiveness, it introduces significant computational overhead due to iterative refinement.

5 Experiments

We evaluate and benchmark our model on QH9 in Section 5.1 and MD17 in Section 5.2. Moreover, we provide the efficiency studies in Section 5.3 and ablation studies in Section 5.4 of SO(2) TP and SO(2) FFNs, demonstrating the contributions of each proposed component. We notice that a new dataset, PubChemQH [13], has recently been introduced with larger molecule size and orbital

Table 1: Performance on QH9 dataset. The unit for the Hamiltonian \mathbf{H} and eigen energies ϵ is Hartree denoted by E_h . Lower is better for \mathbf{H} and ϵ ; higher is better for ψ . The best performance scores are highlighted in **bold**. Underline indicates the second best performance scores.

Dataset	Model	\mathbf{H} [$10^{-6} E_h$] ↓			ϵ [$10^{-6} E_h$] ↓	ψ [10^{-2}] ↑
		diagonal	off-diagonal	all		
QH9-stable-id	QHNet	111.21	73.68	76.31	798.51	95.85
	WANet	–	–	79.99	833.61	96.86
	SPHNet	–	–	45.48	334.28	<u>97.75</u>
	QHNetV2	73.62	28.30	31.50	<u>417.89</u>	98.58
QH9-stable-ood	QHNet	111.72	69.88	72.11	644.17	93.68
	SPHNet	–	–	43.33	186.40	98.16
	QHNetV2	61.09	20.81	22.97	165.89	<u>97.68</u>
QH9-dynamic-300k-geo	QHNet	166.99	95.25	100.19	843.14	94.95
	QHNetV2	87.98	31.67	35.60	270.02	98.77
QH9-dynamic-300k-mol	QHNet	108.70	261.63	119.66	2178.15	90.72
	QHNetV2	138.26	42.06	49.01	629.64	97.43

size. However, as this dataset is not yet publicly available, we will leave the experiments on this dataset until access becomes open. Our code implementation is based on PyTorch 2.4.1 [51], PyTorch Geometric 2.6.1 [52], and e3nn [53]. In experiments, we train models on 80GB Nvidia A100 with Intel(R) Xeon(R) Gold 6258R CPU @ 2.70GHz or 46GB NVIDIA RTX 6000 with AMD EPYC 9684X 96-Core Processor for QH9 and 11GB Nvidia GeForce RTX 2080Ti GPU with Intel Xeon Gold 6248 CPU for MD17.

Evaluation metrics. The evaluation metrics are the mean absolute error (MAE) on Hamiltonian matrix \mathbf{H} , as well as the eigen energies ϵ and cosine similarity on the wavefunction coefficients denoted as ψ on the occupied orbitals.

5.1 QH9

Datasets. The QH9 dataset [11] (CC BY-NC SA 4.0 license) consists of four distinct tasks. The QH9-stable includes 130k molecules derived from QM9 [54, 55]. The detailed description of QH9 dataset can be found in Appendix C.1.

Training Details. Our model follows the same training settings as the QH9 benchmark, with the corresponding hyperparameters provided in Table 5 in the Appendix C.1.

Results. The experimental results are presented in Table 1, where we can observe a clear improvement in the MAE of the Hamiltonian matrix \mathbf{H} . Compared to SPHNet, our model achieves a reduction in MAE on \mathbf{H} by 33.7% on QH9-stable-id and 46.9% on QH9-stable-ood. The improvement is even larger when compared to QHNet. For the MAE of the ϵ , our model achieves at least 47.6% error reduction compared to QHNet, and reasonable results compared to SPHNet.

5.2 MD17

Datasets. The MD17 datasets [41] (CC BY-NC license) consists of four molecular trajectories for water, ethanol, malondialdehyde, and uracil, respectively. The corresponding number of geometries for each trajectory for train/val/test split is provided in Table 6.

Training Details. The training hyperparameters are shown in Table 7 following the settings and split of previous works.

Results. The results of MD17 is summarized in Table 2. We can find that in ethanol, malondialdehyde and uracil, our model provides a significant better performance on the MAE for \mathbf{H} with at least 42% error reduction. Meanwhile, our model provides a better MAE on ϵ . For the water dataset, as shown in Table 6, it contains only 500 geometries compared to 25,000 geometries for each of the other three molecules. It indicates that our model tends to achieve better performance when the training set includes a larger number of geometries.

Table 2: Performance on MD17 dataset. The unit for the Hamiltonian \mathbf{H} and eigen energies ϵ is Hartree denoted by E_h . Lower is better for \mathbf{H} and ϵ ; higher is better for ψ . The best performance scores are highlighted in **bold**. Underline indicates the second best performance scores.

Dataset	Method	Training Strategies	$\mathbf{H}[10^{-6}E_h] \downarrow$	$\epsilon[10^{-6}E_h] \downarrow$	$\psi[10^{-2}] \uparrow$
Water	PhiSNet	LSW (10,000, 200,000)	17.59	<u>85.53</u>	100.00
	QHNet	LSW (10,000, 200,000)	10.36	36.21	99.99
	SPHNet	LSW (10,000, 200,000)	23.18	182.29	100.0
	QHNetV2	LSW (10,000, 200,000)	22.55	106.64	99.99
Ethanol	PhiSNet	LSW (10,000, 200,000)	<u>20.09</u>	102.04	99.81
	QHNet	LSW (10,000, 200,000)	20.91	<u>81.03</u>	99.99
	SPHNet	LSW (10,000, 200,000)	21.02	82.30	100.00
	QHNetV2	LSW (10,000, 200,000)	12.05	70.46	99.99
Malondialdehyde	PhiSNet	LSW (10,000, 200,000)	21.31	100.60	99.89
	QHNet	LSW (10,000, 200,000)	21.52	<u>82.12</u>	99.92
	SPHNet	LSW (10,000, 200,000)	<u>20.67</u>	95.77	99.99
	QHNetV2	LSW (10,000, 200,000)	10.85	67.46	99.92
Uracil	PhiSNet	LSW (10,000, 200,000)	<u>18.65</u>	143.36	99.86
	QHNet	LSW (10,000, 200,000)	20.12	<u>113.44</u>	99.89
	SPHNet	LSW (10,000, 200,000)	19.36	118.21	99.99
	QHNetV2	LSW (10,000, 200,000)	10.38	107.42	99.91

Table 3: Efficiency Studies. This efficiency comparison is performed on the QH9-stable-id task.

Model	Memory [GB/Sample] \downarrow	Speed [Sample/Sec] \uparrow	Speedup Ratio \uparrow
QHNet	0.70	19.20	1.00x
SPHNET	0.23	<u>76.80</u>	<u>4.00x</u>
QHNetV2	<u>0.32</u>	83.33	4.34x

5.3 Efficiency Studies

To compare the efficiency of our model compared to existing baselines, we follow the settings of [14] to run our model on a single A6000 with the maximum available batch size, and compare the corresponding results are shown in Table 3. We can observe a clear speed improvement compared to previous QHNet model with 4.34x faster, demonstrating the efficiency improvement introduced by the new SO(2) operations to eliminate the SO(3) TP operations. Compared to the SPHNet which is based on sparse gate to prune the path in tensor product for accelerating the training procedure, our model shows a faster training speed with a slightly higher GPU memory occupation.

5.4 Ablation Studies

We performed ablation studies on two main components of our model: SO(2) TP and SO(2) FFN. The SO(2) TP is applied to update the node features, while the SO(2) FFN is applied to update the pair node features. The final results are shown in Table 4. We observe that both proposed SO(2) TP and SO(2) FFNs improve the final model performance.

6 Conclusion

In this work, we propose a novel network QHNetV2 for Hamiltonian matrix prediction that leverages the SO(2) local frame to achieve global SO(3) equivariance while eliminating the need for tensor product (TP) operations in both diagonal and off-diagonal components. By introducing the SO(2) local frame, we develop a set of new SO(2)-equivariant operations to construct powerful and efficient neural networks. The proposed approach ensures high computational efficiency by avoiding costly SO(3) TPs entirely. Experimental results on the QH9 and MD17 datasets demonstrate the effectiveness of our method. Our work opens the door to further incorporating efficient and powerful SO(2)-based operations and frames into geometric deep learning, while preserving global SO(3) equivariance.

Table 4: Ablation study results on the QH9-stable-id task for our model with SO(2) TPs on the node updating modules and SO(2) FFNs on the off-diagonal feature updating modules.

Index	Architecture		$H [10^{-6} E_h] \downarrow$	$\epsilon [10^{-6} E_h] \downarrow$	$\psi [10^{-2}] \uparrow$
	SO(2) TP	SO(2) FFNs			
1	✓	✓	31.50	417.89	98.58
2	✗	✓	36.50	429.17	98.18
3	✗	✗	49.84	823.93	97.88

Acknowledgments

SJ acknowledges support from the National Institutes of Health under grant U01AG070112 and ARPA-H under grant 1AY1AX000053. XQ acknowledges support from the National Science Foundation under grant CMMI-2226908 and partial support by the donors of ACS Petroleum Research Fund under grant #65502-ND10.

References

- [1] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136:B864–B871, 1964.
- [2] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140:A1133–A1138, 1965.
- [3] W. Kohn. Nobel lecture: Electronic structure of matter—wave functions and density functionals. *Reviews of Modern Physics*, 71:1253–1266, 1999.
- [4] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N. Kinch, R. Dustin Schaeffer, Claudia Millán, Hahnbeom Park, Carson Adams, Caleb R. Glassman, Andy DeGiovanni, Jose H. Pereira, Andria V. Rodrigues, Alberdina A. van Dijk, Ana C. Ebrecht, Diederik J. Opperman, Theo Sagmeister, Christoph Buhlheller, Tea Pavkov-Keller, Manoj K. Rathinaswamy, Udit Dalwadi, Calvin K. Yip, John E. Burke, K. Christopher Garcia, Nick V. Grishin, Paul D. Adams, Randy J. Read, and David Baker. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.
- [5] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.
- [6] Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, Anima Anandkumar, Karianne Bergen, Carla P. Gomes, Shirley Ho, Pushmeet Kohli, Joan Lasenby, Jure Leskovec, Tie-Yan Liu, Arjun Manrai, Debora Marks, Bharath Ramsundar, Le Song, Jimeng Sun, Jian Tang, Petar Veličković, Max Welling, Linfeng Zhang, Connor W. Coley, Yoshua Bengio, and Marinka Zitnik. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023.
- [7] Xuan Zhang, Limei Wang, Jacob Helwig, Youzhi Luo, Cong Fu, Yaochen Xie, ..., and Shuiwang Ji. Artificial intelligence for science in quantum, atomistic, and continuum systems. *arXiv preprint arXiv:2307.08423*, 2023.
- [8] Oliver Unke, Mihail Bogojeski, Michael Gastegger, Mario Geiger, Tess Smidt, and Klaus-Robert Müller. SE(3)-equivariant prediction of molecular wavefunctions and electronic densities. *Advances in Neural Information Processing Systems*, 34:14434–14447, 2021.

- [9] He Li, Zun Wang, Nianlong Zou, Meng Ye, Runzhang Xu, Xiaoxun Gong, Wenhui Duan, and Yong Xu. Deep-learning density functional theory Hamiltonian for efficient *ab initio* electronic-structure calculation. *Nature Computational Science*, 2(6):367–377, 2022.
- [10] Haiyang Yu, Zhao Xu, Xiaofeng Qian, Xiaoning Qian, and Shuiwang Ji. Efficient and equivariant graph networks for predicting quantum Hamiltonian. In *Proceedings of the 40th International Conference on Machine Learning*, pages 40412–40424, 2023.
- [11] Haiyang Yu, Meng Liu, Youzhi Luo, Alex Strasser, Xiaofeng Qian, Xiaoning Qian, and Shuiwang Ji. QH9: A quantum Hamiltonian prediction benchmark for QM9 molecules. In *The 37th Annual Conference on Neural Information Processing Systems (Track on Datasets and Benchmarks)*, pages 40487–40503, 2023.
- [12] Yang Zhong, Hongyu Yu, Mao Su, Xingao Gong, and Hongjun Xiang. Transferable equivariant graph neural networks for the Hamiltonians of molecules and solids. *npj Computational Materials*, 9(1), 2023.
- [13] Yunyang Li, Zaishuo Xia, Lin Huang, Xinran Wei, Han Yang, Sam Harshe, Zun Wang, Chang Liu, Jia Zhang, Bin Shao, et al. Enhancing the scalability and applicability of Kohn-Sham Hamiltonians for molecular systems. *arXiv preprint arXiv:2502.19227*, 2025.
- [14] Erpai Luo, Xinran Wei, Lin Huang, Yunyang Li, Han Yang, Zun Wang, Chang Liu, Zaishuo Xia, Jia Zhang, and Bin Shao. Efficient and scalable density functional theory Hamiltonian prediction through adaptive sparsity. *arXiv preprint arXiv:2502.01171*, 2025.
- [15] Zhanghao Zhouyin, Zixi Gan, Shishir Kumar Pandey, Linfeng Zhang, and Qiangqiang Gu. Learning local equivariant representations for quantum operators. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [16] Ilyes Batatia, David Peter Kovacs, Gregor N. C. Simm, Christoph Ortner, and Gabor Csanyi. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. In *Advances in Neural Information Processing Systems*, 2022.
- [17] Attila Szabo and Neil S Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Courier Corporation, 2012.
- [18] Kristof T Schütt, Huziel E Sauceda, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller. SchNet – a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 2018.
- [19] Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2020.
- [20] Yi Liu, Limei Wang, Meng Liu, Yuchao Lin, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3D molecular graphs. In *International Conference on Learning Representations*, 2022.
- [21] Limei Wang, Yi Liu, Yuchao Lin, Haoran Liu, and Shuiwang Ji. ComENet: Towards complete and efficient message passing for 3D molecular graphs. In *The 36th Annual Conference on Neural Information Processing Systems*, pages 650–664, 2022.
- [22] Guillem Simeon and Gianni De Fabritiis. TensorNet: Cartesian tensor representations for efficient learning of molecular potentials. *Advances in Neural Information Processing Systems*, 36:37334–37353, 2023.
- [23] Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J Bekkers, and Max Welling. Geometric and physical quantities improve E(3) equivariant message passing. In *International Conference on Learning Representations*, 2022.
- [24] Yuanqing Wang and John Chodera. Spatial attention kinetic networks with E(n)-equivariance. In *The Eleventh International Conference on Learning Representations*, 2023.
- [25] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3D point clouds. *arXiv Preprint, arXiv:1802.08219*, 2018.

- [26] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. SE(3)-transformers: 3D roto-translation equivariant attention networks. *Advances in Neural Information Processing Systems*, 33:1970–1981, 2020.
- [27] Albert Musaelian, Simon Batzner, Anders Johansson, Lixin Sun, Cameron J Owen, Mordechai Kornbluth, and Boris Kozinsky. Learning local equivariant representations for large-scale atomistic dynamics. *Nature Communications*, 14(1):579, 2023.
- [28] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13(1):2453, 2022.
- [29] Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant graph attention transformer for 3D atomistic graphs. In *The Eleventh International Conference on Learning Representations*, 2023.
- [30] Klaus-Robert Müller, Oliver Thorsten Unke, and Thorben Frank. So3krates - self-attention for higher-order geometric interactions on arbitrary length-scales. 2022.
- [31] Victor Garcia Satorras, Emiel Hooeboom, and Max Welling. E(n) equivariant graph neural networks. In *International Conference on Machine Learning*, pages 9323–9332. PMLR, 2021.
- [32] Shengjie Luo, Tianlang Chen, and Aditi S Krishnapriyan. Enabling efficient equivariant operations in the Fourier basis via Gaunt Tensor Products. In *The Twelfth International Conference on Learning Representations*.
- [33] Zhao Xu, Haiyang Yu, Montgomery Bohde, and Shuiwang Ji. Equivariant graph network approximations of high-degree polynomials for force field prediction. *Transactions on Machine Learning Research*, 2024.
- [34] Saro Passaro and C Lawrence Zitnick. Reducing SO(3) convolutions to SO(2) for efficient equivariant gnns. In *International Conference on Machine Learning*, pages 27420–27438. PMLR, 2023.
- [35] Yuchao Lin, Jacob Helwig, Shurui Gui, and Shuiwang Ji. Equivariance via minimal frame averaging for more symmetries and efficiency. In *Proceedings of the 41st International Conference on Machine Learning*, pages 30042–30079, 2024.
- [36] Omri Puny, Matan Atzmon, Heli Ben-Hamu, Ishan Misra, Aditya Grover, Edward J Smith, and Yaron Lipman. Frame averaging for invariant and equivariant network design. *arXiv preprint arXiv:2110.03336*, 2021.
- [37] Nadav Dym, Hannah Lawrence, and Jonathan W Siegel. Equivariant frames and the impossibility of continuous canonicalization. *arXiv preprint arXiv:2402.16077*, 2024.
- [38] Yi-Lun Liao, Brandon M Wood, Abhishek Das, and Tess Smidt. EquiformerV2: Improved equivariant transformer for scaling to higher-degree representations. In *The Twelfth International Conference on Learning Representations*, 2024.
- [39] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- [40] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.
- [41] Kristof T Schütt, Michael Gastegger, Alexandre Tkatchenko, K-R Müller, and Reinhard J Maurer. Unifying machine learning and quantum chemistry with a deep neural network for molecular wavefunctions. *Nature Communications*, 10(1):5024, 2019.
- [42] He Li, Zun Wang, Nianlong Zou, Meng Ye, Runzhang Xu, Xiaoxun Gong, Wenhui Duan, and Yong Xu. Deep-learning density functional theory Hamiltonian for efficient *ab initio* electronic-structure calculation. *Nature Computational Science*, 2(6):367–377, 2022.

- [43] Xiaoxun Gong, He Li, Nianlong Zou, Runzhang Xu, Wenhui Duan, and Yong Xu. General framework for E(3)-equivariant neural network representation of density functional theory Hamiltonian. *Nature Communications*, 14(1):2848, 2023.
- [44] Yuxiang Wang, He Li, Zechen Tang, Honggeng Tao, Yanzhen Wang, Zilong Yuan, Zezhou Chen, Wenhui Duan, and Yong Xu. DeepH-2: enhancing deep-learning electronic structure via an equivariant local-coordinate transformer. *arXiv preprint arXiv:2401.17015*, 2024.
- [45] He Li, Zechen Tang, Xiaoxun Gong, Nianlong Zou, Wenhui Duan, and Yong Xu. Deep-learning electronic-structure calculation of magnetic superstructures. *Nature Computational Science*, 3(4):321–327, 2023.
- [46] Zilong Yuan, Zechen Tang, Honggeng Tao, Xiaoxun Gong, Zezhou Chen, Yuxiang Wang, He Li, Yang Li, Zhiming Xu, Minghui Sun, et al. Deep learning density functional theory Hamiltonian in real space. *arXiv preprint arXiv:2407.14379*, 2024.
- [47] Shi Yin, Xinyang Pan, Fengyan Wang, and Lixin He. TraceGrad: a framework learning expressive SO(3)-equivariant non-linear representations for electronic-structure Hamiltonian prediction. *arXiv preprint arXiv:2405.05722*, 2024.
- [48] Feitong Song and Ji Feng. NeuralSCF: Neural network self-consistent fields for density functional theory. *arXiv preprint arXiv:2406.15873*, 2024.
- [49] He Zhang, Chang Liu, Zun Wang, Xinran Wei, Siyuan Liu, Nanning Zheng, Bin Shao, and Tie-Yan Liu. Self-consistency training for density-functional-theory Hamiltonian prediction. In *International Conference on Machine Learning*, pages 59329–59357. PMLR, 2024.
- [50] Alexander Mathiasen, Hatem Helal, Paul Balanca, Adam Krzywaniak, Ali Parviz, Frederik Hvilshøj, Blazej Banaszewski, Carlo Luschi, and Andrew William Fitzgibbon. Reducing the cost of quantum chemical data by backpropagating through density functional theory. *arXiv preprint arXiv:2402.04030*, 2024.
- [51] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.
- [52] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with PyTorch Geometric. *arXiv Preprint, arXiv:1903.02428*, 2019.
- [53] Mario Geiger, Tess Smidt, Alby M., Benjamin Kurt Miller, Wouter Boomsma, Bradley Dice, Kostiantyn Lapchevskyi, Maurice Weiler, Michał Tyszkiewicz, Simon Batzner, Dylan Madiseti, Martin Uhrin, Jes Frellsen, Nuri Jung, Sophia Sanborn, Mingjian Wen, Josh Rackers, Marcel Rød, and Michael Bailey. Euclidean neural networks: e3nn, 2022.
- [54] Lars Ruddigkeit, Ruud Van Deursen, Lorenz C Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *Journal of Chemical Information and Modeling*, 52(11):2864–2875, 2012.
- [55] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1(1):140022, 2014.

A Background

A.1 Density Functional Theory

Density Functional Theory (DFT) [1, 2, 3] provides a powerful and efficient computational framework for modeling electronic structures of quantum systems and predicting a wide range of chemical and physical properties for gas-phase molecules and solid-state materials. DFT-based electronic structure methods reduce the reliance on the laboratory experiments, significantly advancing scientific research in physics, chemistry, and materials science. The key motivation behind DFT is to address the well-known many-body Schrödinger equation that governs the quantum states of systems such as their energies and wavefunctions. Mathematically, the Schrödinger equation for a system of total N_e electrons is given by: $\hat{H}\Psi(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_{N_e}) = E\Psi(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_{N_e})$, where \hat{H} is the Hamiltonian operator and $\Psi(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_{N_e})$ is the many-body electronic wavefunction. Although the Schrödinger equation can describe the entire system exactly, its computational complexity grows exponentially with the number of electrons. Specifically, even without accounting for the spin degrees of freedom, the many-body electronic wave function Ψ depends on $3N_e$ -dimensional spatial variables. Hence, the associated function space expands exponentially with respect to N_e , making it computationally infeasible for computers to accurately solve complex and diverse quantum systems. To address this challenge, Hohenberg and Kohn proved that the ground-state properties of a many-body electronic system is uniquely determined by its three-dimensional electron density $\rho(\mathbf{r})$ [1], thus avoiding the need to explicitly handle the full $3N_e$ -dimensional many-electron wavefunction $\Psi(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_{N_e})$. Kohn and Sham proposed a practical approach to map a many-body interacting system onto a set of non-interacting one-body systems where each electron moves in an effective potential arising from the nuclei and the average effect of other electrons [2]. This leads to the well-known single-particle Kohn-Sham equation: $\hat{H}_{\text{KS}}|\psi_p\rangle = \epsilon_p|\psi_p\rangle$ for individual electrons, where \hat{H}_{KS} is the Kohn-Sham Hamiltonian, ψ_p is the single-electron eigen wavefunction, and ϵ_p is the corresponding eigen energy. The Kohn-Sham Hamiltonian is given by $\hat{H}_{\text{KS}} = \hat{T} + \hat{V}_{\text{Hartree}} + \hat{V}_{\text{XC}} + \hat{V}_{\text{ext}}$, where $\hat{T} = -\frac{\hbar^2}{2m_e}\nabla^2$ is the kinetic energy operator, \hat{V}_{Hartree} is the Hartree potential, \hat{V}_{XC} is the exchange-correlation potential, and \hat{V}_{ext} is external potential. The electron density from the Kohn-Sham equation with the exact exchange-correlation energy functional, should be same as that of the many-body interacting system: $\rho(\mathbf{r}) = \sum_p \int f_p |\psi_p(\mathbf{r})|^2 d\mathbf{r} = \int |\Psi(\mathbf{r}, \mathbf{r}_2, \dots, \mathbf{r}_{N_e})|^2 d\mathbf{r}_2 \dots d\mathbf{r}_{N_e}$, where f_p is the occupation number of the p -th Kohn-Sham eigen state. Therefore, solving the full many-electron wavefunction Ψ is no longer required.

A.2 Atomic Orbitals and Hamiltonian Matrices

The Kohn-Sham equation, as the central equation in DFT, can be solved in a predefined basis set, such as the STO-3G atomic orbital basis which combines radial functions with spherical harmonics centered on each atom. The calculated Kohn-Sham eigen wavefunction ψ_n , often referred to as molecular orbital in the context of molecules, can be expressed as a linear combination of these predefined basis functions $\psi_n(\mathbf{r}) = \sum_p \mathbf{C}_{pn} \phi_p(\mathbf{r})$, known as the Linear Combination of Atomic Orbital (LCAO) method [17]. Each single-electron wavefunction $\psi_n(\mathbf{r})$ depends only on three spatial variables, thereby avoiding the exponential scaling associated with the $3N_e$ -dimensional full many-electron wavefunction. By applying the predefined orbitals and the LCAO method within DFT, the Kohn-Sham equation can be converted into the following matrix form:

$$\mathbf{H}\mathbf{C} = \mathbf{\epsilon}\mathbf{S}\mathbf{C}, \quad (15)$$

where \mathbf{H} is the Hamiltonian matrix of size $\mathbb{R}^{N_o \times N_o}$. Each matrix element is defined by evaluating the Hamiltonian operator in a pair of predefined atomic orbitals, shown as $\mathbf{H}_{pq} \equiv \langle \phi_p | \hat{H}_{\text{KS}} | \phi_q \rangle = \int \phi_p^*(\mathbf{r}) \hat{H}_{\text{KS}} \phi_q(\mathbf{r}) d\mathbf{r}$. Here, N_o denotes the number of predefined orbitals which typically increases linearly with the number of electronics N_e . $\mathbf{C} \in \mathbb{R}^{N_o \times N_o}$ denotes the wavefunction coefficient matrix. $\mathbf{\epsilon} \in \mathbb{R}^{N_o \times N_o}$ is a diagonal matrix where each diagonal element corresponds to the eigen energy of each molecular orbital. $\mathbf{S} \in \mathbb{R}^{N_o \times N_o}$ is the overlap matrix, where each entry is the integral of a pair of predefined basis over the spatial space $\mathbf{S}_{pq} = \int \phi_p^*(\mathbf{r}) \phi_q(\mathbf{r}) d\mathbf{r}$.

A.3 SO(3) Equivariance of Hamiltonian Matrices

Equivariance is a fundamental symmetry that must be preserved in the Hamiltonian matrix prediction task. It arises from representing Hamiltonian matrix and wavefunctions in specific basis sets such as atomic orbitals in DFT calculations which are sensitive to the spatial orientation. As a result, when a molecular system undergoes a global rotation characterized by Euler angles (α, β, γ) , its Hamiltonian matrix must transform accordingly. Each atom pairs and their orbital pairs can be labeled as (a_i, o_s, a_j, o_t) . a_i and a_j are the atom indices which may refer to the same atom ($i = j$) or to different atoms ($i \neq j$) in the molecular system. o_s and o_t indicate the orbitals belonging to a_i and a_j , respectively, with angular momentum quantum numbers ℓ_s and ℓ_t and magnetic quantum number m_s and m_t . Under a rotation specified by Euler angles (α, β, γ) , the corresponding Hamiltonian matrix block transforms as $\mathbf{H}'_{ij,st} = (D^{\ell_s}(\alpha, \beta, \gamma))^{-1} \mathbf{H}_{ij,st} D^{\ell_t}(\alpha, \beta, \gamma)$, where two Wigner-D matrices $D^\ell(\alpha, \beta, \gamma)$ are applied to the left and right sides of the original Hamiltonian matrix block, respectively, according to the angular momentum of the orbital pairs in the block. This ensures the block-wise SO(3) equivariance of the Hamiltonian matrix under rotations.

B SO(3) Equivariance of SO(2) Local Frames

Frame averaging [36, 37, 35] is a technique that enforces equivariance to any model Φ by an SO(3)-equivariant frame $\mathcal{F} : \mathbb{R}^3 \rightarrow \text{SO}(3)$ such that

$$\langle \Phi \rangle_{\mathcal{F}(\hat{r})} = \frac{1}{|\mathcal{F}(\hat{r})|} \sum_{g \in \mathcal{F}(\hat{r})} g \cdot \Phi(g^{-1} \cdot \hat{r}) \quad (16)$$

With a little abuse of the definition, we further consider the input space includes the SO(3) irreps which is build upon the global coordinate system. Since there are many candidate frame constructions for frame averaging, we use the minimal frame averaging [35] to give an analytic frame construction with a minimal frame size. To apply the minimal frame averaging [35], the canonicalization is required to transfer the input features into its canonical form, defined in Definition 3.3 [35]. We define the canonicalization operation based on a reference vector \hat{r} as $c(\hat{r}) = h^{-1} \cdot \hat{r} = \hat{v}$, where a rotation $h \in \text{SO}(3)$ is applied to align the reference vector \hat{r} with the fixed target vector \hat{v} .

By Theorem 3.2 in [35], the minimal frame is defined by $\mathcal{F}(\hat{r}) = h \text{Stab}_{\text{SO}(3)}(\hat{v})$ where

$$\text{Stab}_{\text{SO}(3)}(\hat{v}) = \{g \in \text{SO}(3) \mid g \cdot \hat{v} = \hat{v}\}.$$

Instead of employing an arbitrary local model which could make the direct sum over all $g \in \text{Stab}_{\text{SO}(3)}(\hat{v})$ computationally expensive or even intractable, since $\text{Stab}_{\text{SO}(3)}(\hat{v})$ is isomorphic to SO(2) (rotations about the y-axis), we consider Φ to be $\text{Stab}_{\text{SO}(3)}(\hat{v})$ -equivariant with respect to the reference axis r . That's to say, for any rotation $g \in \text{Stab}_{\text{SO}(3)}(\hat{v})$, the local network satisfies $\Phi(g \cdot \hat{r}) = g \Phi(\hat{r})$. In this case, the frame-averaging sum collapses such that

$$\begin{aligned} \langle \Phi \rangle_{\mathcal{F}(\hat{r})} &= \frac{1}{|\mathcal{F}(\hat{r})|} \sum_{g \in \mathcal{F}(\hat{r})} g \cdot \Phi(g^{-1} \hat{r}) \\ &= \frac{1}{|\text{Stab}_{\text{SO}(3)}(\hat{v})|} \sum_{g \in \text{Stab}_{\text{SO}(3)}(\hat{v})} hg \cdot \Phi((hg)^{-1} \cdot \hat{r}) \\ &= \frac{1}{|\text{Stab}_{\text{SO}(3)}(\hat{v})|} \sum_{g \in \text{Stab}_{\text{SO}(3)}(\hat{v})} hg \cdot \Phi(g^{-1} h^{-1} \cdot \hat{r}) \\ &= \frac{1}{|\text{Stab}_{\text{SO}(3)}(\hat{v})|} \sum_{g \in \text{Stab}_{\text{SO}(3)}(\hat{v})} hgg^{-1} \cdot \Phi(h^{-1} \cdot \hat{r}) \\ &= \frac{1}{|\text{Stab}_{\text{SO}(3)}(\hat{v})|} \sum_{g \in \text{Stab}_{\text{SO}(3)}(\hat{v})} h \cdot \Phi(h^{-1} \cdot \hat{r}) \\ &= h \cdot \Phi(h^{-1} \cdot \hat{r}). \end{aligned} \quad (17)$$

Thus, the entire frame reduces to the single rotation h making the computation simple and tractable. We then define the \mathcal{F} built upon the $\text{Stab}_{\text{SO}(3)}(r)$ -equivariant Φ as the *SO(2) local frame*.

C Hyperparameter settings and model architectures

C.1 QH9

In the QH9-stable-id setting, the training, validation and test sets are randomly split. In contrast, the QH9-stable-ood settings use a split based on molecular size, with validation and test sets consisting of molecules larger than those in the training set. The QH9-dynamic-300k dataset is an extended version of the QH9-dynamic benchmark, comprising 300k molecular geometries generated from 3k molecules, each with a molecular dynamics trajectory of 100 steps. In the QH9-dynamic-300k-geo setting, the training, validation, and test sets consist of the same molecules but different geometries sampled from their respective trajectories. In contrast, the QH9-dynamic-300k-mol setting is split based on different molecules, with all geometries from a given trajectory assigned to the same split. This setup is designed to evaluate the model’s ability to generalize to unseen molecules with diverse geometries. We follow the model and training hyperparameters in Table 5 to perform our experiments.

Table 5: Hyperparameters in QH9.

Hyperparameters	Values/search space
Batch size	32
Cutoff distance (Bohr)	15
Initial learning rate	5e-4
Final learning rate	1e-7
Learning rate strategy	linear schedule with warmup
Learning rate warmup batches	1,000
# of batches in training	26,000
# of layers	3
L_{max}	4
hidden irreps	256x0e+128x1e+64x2e+32x3e+16x4e
hidden mirreps	1024x0m+256x1m+64x2m+32x3m+16x4m
hidden ffn mirreps	2048x0m+512x1m+256x2m+64x3m+32x4m

C.2 MD17

The statistics of MD17 is shown in Table 6. We follow the model and training hyperparameters in Table 7 to perform our experiments. Specifically, for the model molecule, we choose a smaller model since the number of training geometries is smaller. We choose hidden irreps to be $64 \times 0e + 32 \times 1e + 16 \times 2e + 8 \times 3e + 8 \times 4e$, hidden mirreps to be $64 \times 0m + 32 \times 1m + 16 \times 2m + 8 \times 3m + 8 \times 4m$, and hidden ffn mirreps to be $64 \times 0m + 32 \times 1m + 16 \times 2m + 8 \times 3m + 8 \times 4m$.

Table 6: The statistics of MD17 dataset.

Dataset	Molecule	# of structures	Train	Val	Test
Water	H ₂ O	4,900	500	500	3,900
Ethanol	C ₂ H ₅ OH	30,000	25,000	500	4,500
Malondialdehyde	CH ₂ (CHO) ₂	26,978	25,000	500	1,478
Uracil	C ₄ H ₄ N ₂ O ₂	30,000	25,000	500	4,500

Table 7: Hyperparameters in MD17.

Hyperparameters	Values/search space
Batch size	5, 10, 16
Cutoff distance (Bohr)	15
Initial learning rate	1e-3, 5e-4
Final learning rate	1e-6, 1e-7
Learning rate strategy	linear schedule with warmup
Learning rate warmup batches	1,000
# of batches in training	20,000
# of layers	2, 3
L_{max}	4
hidden irreps	256x0e+128x1e+64x2e+32x3e+16x4e
hidden mirreps	1024x0m+256x1m+64x2m+32x3m+16x4m
hidden ffn mirreps	2048x0m+512x1m+256x2m+64x3m+32x4m