

EFFICIENT PREDICTION OF $\text{SO}(3)$ -EQUIVARIANT HAMILTONIAN MATRICES VIA $\text{SO}(2)$ LOCAL FRAMES

005 **Anonymous authors**

006 Paper under double-blind review

ABSTRACT

011 We consider the task of predicting Hamiltonian matrices to accelerate electronic
 012 structure calculations, which plays an important role in physics, chemistry, and
 013 materials science. Motivated by the inherent relationship between the off-diagonal
 014 blocks of the Hamiltonian matrix and the $\text{SO}(2)$ local frame, we propose a novel
 015 and efficient network, called QHNetV2, that achieves global $\text{SO}(3)$ equivariance
 016 without the costly $\text{SO}(3)$ Clebsch–Gordan tensor products. This is achieved by
 017 introducing a set of new efficient and powerful $\text{SO}(2)$ -equivariant operations and
 018 performing all off-diagonal feature updates and message passing within $\text{SO}(2)$
 019 local frames, thereby eliminating the need of $\text{SO}(3)$ tensor products. Moreover, a
 020 continuous $\text{SO}(2)$ tensor product is performed within the $\text{SO}(2)$ local frame at each
 021 node to fuse node features. Extensive experiments on the large QH9 and MD17
 022 datasets demonstrate that our model achieves superior performance across a wide
 023 range of molecular structures and trajectories, highlighting its strong generalization
 024 capability. The proposed $\text{SO}(2)$ operations on $\text{SO}(2)$ local frames offer a promising
 025 direction for scalable and symmetry-aware learning of electronic structures.

1 INTRODUCTION

029 Quantum Hamiltonian, as a central element in the many-body Schrödinger equation of quantum
 030 mechanics, plays a key role in governing the quantum states and physical properties of molecules and
 031 materials, making it essential for physics, chemistry, and materials science. First-principles methods
 032 such as density functional theory (DFT) (1; 2; 3) have been developed to solve the Schrödinger
 033 equation and investigate the electronic structures of molecules and solids. Despite their success, these
 034 methods struggle with high computational cost, limiting their application to systems with only a few
 035 hundred atoms. In DFT, the central Kohn-Sham equation is solved using the self-consistent field (SCF)
 036 method based on the variational principle of the second Hohenberg-Kohn theorem (2). The electronic
 037 wavefunctions, energies, charge density, and Kohn-Sham Hamiltonian are iteratively calculated and
 038 updated until convergence is achieved. This SCF calculation has a high time complexity of $O(N_e^3 T)$,
 039 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 B.1.

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

047 To address the above challenge, here we propose a novel and efficient network QHNetV2, motivated
 048 by the inherent relationship between the off-diagonal blocks of the Hamiltonian matrix and the $\text{SO}(2)$
 049 local frame to achieve global $\text{SO}(3)$ equivariance without using the computationally intensive $\text{SO}(3)$
 050 Clebsch–Gordan tensor products. Specifically, we first eliminate the need of $\text{SO}(3)$ tensor products
 051 by introducing a set of new efficient and powerful $\text{SO}(2)$ -equivariant operations and performing all
 052 off-diagonal feature updates and message passing within $\text{SO}(2)$ local frames. Second, to effectively
 053 perform nonlinear node updates, we apply a continuous $\text{SO}(2)$ tensor product within the $\text{SO}(2)$ local
 frame at each node to fuse node features, mimicking the symmetric contraction module used in

054 MACE (16) for modeling many-body interactions. We conduct extensive experiments on the large
 055 QH9 and MD17 datasets, which shows the superior performance and strong generalization capability
 056 of our new framework over a diverse set of molecular structures and trajectories. Additionally,
 057 these novel SO(2) operations on SO(2) local frames presents a promising avenue for scalable and
 058 symmetry-aware learning of electronic structures.

060 2 PRELIMINARIES

062 2.1 HAMILTONIAN MATRICES AND SO(3) EQUIVARIANCE

064 The Kohn-Sham equation of molecular and materials systems is given by: $\hat{H}_{\text{KS}}|\psi_n\rangle = \epsilon_n|\psi_n\rangle$,
 065 where \hat{H}_{KS} is the Kohn-Sham Hamiltonian operator, ψ_n is single-electron eigen wavefunction (also
 066 called molecular orbital), and ϵ_n is the corresponding eigen energy. Under a predefined basis set such
 067 as the STO-3G atomic orbital basis $\{\phi_p\}$ combining radial functions with spherical harmonics, the
 068 Kohn-Sham equation can be converted into a matrix form: $\mathbf{HC} = \epsilon \mathbf{SC}$, with Hamiltonian matrix
 069 element $\mathbf{H}_{pq} = \int \phi_p^*(\mathbf{r}) \hat{H}_{\text{KS}} \phi_q(\mathbf{r}) d\mathbf{r}$ and overlap matrix element $\mathbf{S}_{pq} = \int \phi_p^*(\mathbf{r}) \phi_q(\mathbf{r}) d\mathbf{r}$. ϵ is a
 070 diagonal matrix of the eigen energies, and \mathbf{C} contains wavefunction coefficients, with each eigen
 071 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). More
 072 details can be found in Appendices B.1, B.2.

073 Equivariance is a fundamental symmetry that must be preserved in the Hamiltonian matrix prediction
 074 task. It arises from representing Hamiltonian matrix and wavefunctions in specific basis sets such
 075 as atomic orbitals in DFT calculations which are sensitive to the spatial orientation. As a result,
 076 when a molecular system undergoes a global rotation characterized by Euler angles (α, β, γ) , its
 077 Hamiltonian matrix must transform accordingly. Each atom pairs and their orbital pairs can be labeled
 078 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
 079 different atoms ($i \neq j$) in the molecular system. o_s and o_t indicate the orbitals belonging to a_i and
 080 a_j , respectively, with angular momentum quantum numbers ℓ_s and ℓ_t and magnetic quantum number
 081 m_s and m_t . Under a rotation specified by Euler angles (α, β, γ) , the corresponding Hamiltonian
 082 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
 083 matrices $D^\ell(\alpha, \beta, \gamma)$ are applied to the left and right sides of the original Hamiltonian matrix block,
 084 respectively, according to the angular momentum of the orbital pairs in the block. This ensures the
 085 block-wise SO(3) equivariance of the Hamiltonian matrix under rotations.

086 When learning the Hamiltonian matrix using machine learning models, it is essential to incorporate
 087 high-degree equivariant features that align with the angular momentum quantum numbers ℓ of atomic
 088 orbitals. This is particularly important for accurately capturing orbitals with high angular momentum
 089 such as d -orbitals with $\ell = 2$, requiring the highest degree of SO(3) equivariant features with
 090 $L_{\max} \geq 4$. Furthermore, the pairwise interactions are needed for predicting all Hamiltonian matrix
 091 blocks while maintaining the block-wise SO(3) equivariance.

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

104 2.2 SO(3) AND SO(2) IRREDUCIBLE REPRESENTATIONS

105 Under an arbitrary 3D rotation, the atomic orbital basis rotates accordingly. As a result, the Hamil-
 106 tonian matrix element defined on the atomic orbital pairs transforms in an equivariant manner. It
 107 is therefore necessary to introduce irreducible representations (irreps) of the 3D rotation group, *i.e.*

108 SO(3) group. The SO(3) group consists of all 3D rotations, represented by 3×3 orthogonal matrices
 109 with determinant 1. For each SO(3) irrep with degree $\ell \in \mathbb{N}_0$, the corresponding representation space
 110 has dimension $2\ell + 1$, with spherical harmonics $\{Y_m^\ell(\theta, \phi)\}_{m=-\ell}^\ell$ serving as a natural basis. For
 111 each group element $g \in \text{SO}(3)$ (*i.e.* 3D rotation), the corresponding matrix representation for the ℓ -th
 112 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
 113 multiplication to perform the corresponding SO(3) rotation in the representation space. To develop
 114 more advanced operations within the SO(2) symmetry space, it is essential to introduce irreducible
 115 representations (irreps) of the SO(2) group. The SO(2) group describes 2D planar rotations, which
 116 can be interpreted as rotations around a fixed axis in 3D space (typically the z-axis). The irreps
 117 of SO(2) has dimension 2 for $m > 0$ and 1 for $m = 0$ or can be represented as complex values,
 118 with order $m \in \mathbb{N}_0$. A natural basis function for SO(2) irreps is the set of real circular harmonics,
 119 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
 120 group element $g \in \text{SO}(2)$ corresponds to a rotation by an angle $\varphi \in [0, 2\pi)$, and the matrix rep-
 121 resentation 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 $m = 0$.
 122 Subsequently, the representation space under SO(2) rotation transforms via matrix multiplication. In
 123 the complex circular harmonics basis $e^{im\delta}$, the representation space under SO(2) rotation transforms
 124 via multiplication with $e^{im\varphi}$.

3 METHODS

126 This section introduces the model we have built for the Hamiltonian matrix prediction task. The key
 127 is to develop the SO(2) local frame, where any powerful SO(2) equivariant operations can be applied
 128 while maintaining the overall framework to be SO(3) equivariant. Here, we would like to clarify
 129 that we adopt minimal frame averaging and extend the operation in eSCN to construct local frames,
 130 enabling the application of arbitrary SO(2) operations and supporting frame construction on nodes,
 131 edges, and node pairs beyond edges alone. Later on, we demonstrate that the model based on SO(2)
 132 local frame achieve great efficiency and accuracy, especially these two key things for building this
 133 networks especially considering the high L_{\max} is an unavoidable things.

3.1 SO(2) LOCAL FRAMES

137 While the computation cost for SO(3) operations remains high, SO(2) operations can be more
 138 efficient to conduct. Besides the SO(2) linear operation, which has been verified in previous
 139 work (34) to maintain overall SO(3) equivariance, there is a need to provide a mechanism that
 140 supports the usage of arbitrary SO(2) operations while ensuring the framework remains SO(3)-
 141 equivariant. Therefore, inspired by the minimal frame averaging technique (35), we construct local
 142 frames that require SO(2) equivariance internally and employ canonicalization to guarantee overall
 143 SO(3) equivariance.

144 **Global Frame.** The overall framework is built on a global coordinate system together with local
 145 SO(2) frames. In the global system, the model operates on SO(3) irreps and preserves full SO(3)
 146 equivariance. Consequently, the input to each local network Φ is SO(3)-equivariant.

147 **Definition of the Local Frame.** A local frame is defined as a mapping from the 3D Euclidean
 148 space to SO(3):

$$\mathcal{F} : \mathbb{R}^3 \rightarrow \text{SO}(3). \quad (1)$$

149 To make this concrete, we introduce a fixed target vector $\hat{v} \in \mathbb{R}^3$ that is used in the minimal frame
 150 construction. Given a unit direction vector $\hat{r} \in \mathbb{R}^3$, which rotates consistently with the input 3D data,
 151 and the fixed target vector \hat{v} , the local frame $F(\hat{r})$ is the rotation that maps \hat{r} onto \hat{v} . For example,
 152 if we take the node pair direction \hat{r}_{ij} and set $\hat{v} = (0, 0, 1)$, this reduces exactly to the setup used in
 153 eSCN. Thus, eSCN can be viewed as a special case of our more general framework. As proved in
 154 Appendix C, such a local frame can be applied to any SO(2) operation beyond the linear case by
 155 incorporating local frame averaging. Moreover, the placement of SO(2) local frames can be extended
 156 whenever a reference unit vector \hat{r} is available, for example by defining a local frame on each node.

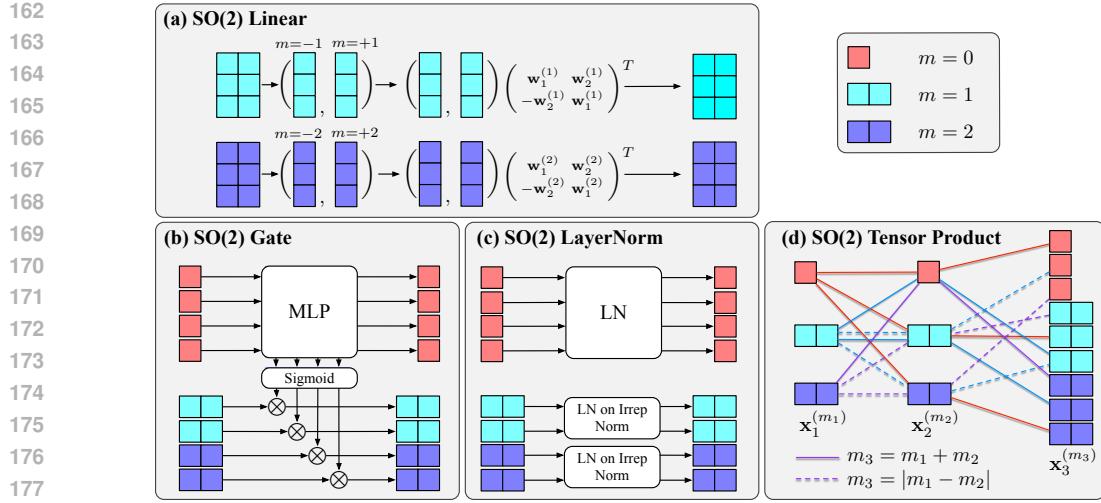


Figure 1: SO(2) equivariant operations. (a) SO(2) Linear. For SO(2) irreps with order $m > 0$, this operation uses weight matrices $\mathbf{w}_1^{(m)}, \mathbf{w}_2^{(m)} \in \mathbb{R}^{C \times C}$ where C is the number of channels for input irreps. (b) SO(2) Gate. For the $m = 0$ features, a multi-layer 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. 6. (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. 9. Each valid combination defines a path, ensuring the resulting features remain SO(2)-equivariant.

Mapping Between SO(3) and SO(2) Irreps. We use $FR(\hat{r})$ to denote the operation that projects SO(3) irreps into SO(2) irreps within the local frame, and $FR(\hat{r})^{-1}$ for the inverse mapping. Formally, given an SO(3) irrep $x \in \mathbb{R}^{2\ell+1}$ with components indexed by $m \in \{-\ell, \dots, \ell\}$, the corresponding SO(2) irrep x' after $FR(\hat{r})$ is defined as

$$x'_m = \sum_{m'=-\ell}^{\ell} D^\ell(R)_{m,m'} x_{m'}, \quad (2)$$

where $D^\ell(R) \in \mathbb{R}^{(2\ell+1) \times (2\ell+1)}$ is the Wigner- D matrix associated with the rotation R obtained from the canonicalization procedure to rotate reference direction \hat{r} onto fixed target vector \hat{v} . The equivariance of such transformation is shown in Appendix E.

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.

SO(2) Linear. SO(2) Linear layer is first introduced in eSCN (34), and used in following works like EquiformerV2 (36). 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

$$\begin{pmatrix} \mathbf{z}_{c,-m} \\ \mathbf{z}_{c,m} \end{pmatrix} = \sum_{c'} \begin{pmatrix} \mathbf{w}_{1,cc'}^{(m)} & \mathbf{w}_{2,cc'}^{(m)} \\ -\mathbf{w}_{2,cc'}^{(m)} & \mathbf{w}_{1,cc'}^{(m)} \end{pmatrix} \begin{pmatrix} \mathbf{x}_{c',-m} \\ \mathbf{x}_{c',m} \end{pmatrix}, \quad (3)$$

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

216 $\tilde{\mathbf{x}}_{c'}^{(m)} = \mathbf{x}_{c',m} + i\mathbf{x}_{c',-m} = \bar{\mathbf{x}}_{c'}^{(m)} e^{i\theta_{x^{(m)},c'}}$ and weights $\tilde{\mathbf{w}}_{cc'}^{(m)} = \mathbf{w}_{1,cc'}^{(m)} + i\mathbf{w}_{2,cc'}^{(m)} = \bar{\mathbf{w}}_{cc'}^{(m)} e^{i\theta_{w^{(m)},cc'}}$.
 217 The above equation is equivalent to
 218

$$219 \quad \tilde{\mathbf{z}}_c^{(m)} = \sum_{c'} \tilde{\mathbf{w}}_{cc'}^{(m)} \tilde{\mathbf{x}}_{c'}^{(m)} = \sum_{c'} \bar{\mathbf{x}}_{c'}^{(m)} \bar{\mathbf{w}}_{cc'}^{(m)} e^{i(\theta_{w^{(m)},cc'} + \theta_{x^{(m)},c'})}, \quad (4)$$

222 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
 223 internal weights on the scale $\bar{\mathbf{w}}^{(m)}$ and rotation with angle $\theta_{w^{(m)}}$, as well as self-interaction across
 224 all input channels. Note that there is a set of weights for each individual \mathbf{x}^m with $m > 0$.
 225

226 **SO(2) Gate.** Gate activation is a useful component on various networks for SO(3) irreducible
 227 representations (28; 36), and this operation can also be applied to the SO(2) irreducible representations.
 228 Specifically, the corresponding formulation is shown as
 229

$$230 \quad \mathbf{z}^{(m)} = \begin{cases} \text{MLP}(\mathbf{x}^{m=0}) & , \text{if } m = 0, \\ \text{Sigmoid}(\text{MLP}(\mathbf{x}^{m=0})) \circ \mathbf{x}^{(m)} & , \text{if } m > 0, \end{cases} \quad (5)$$

231 An MLP is applied to learn the $m = 0$ features, and the other MLP takes the $m = 0$ features as
 232 the input, applies a sigmoid gate activation, then multiplies with the $m > 0$ features to control the
 233 irreducible features.
 234

235 **SO(2) Layer Normalization.** Within the overall framework, it is often necessary to maintain a
 236 set of SO(2)-equivariant features to model the target quantities. In our task, for each off-diagonal
 237 block of the Hamiltonian matrix, it is required to model its corresponding SO(2) features. Since
 238 LayerNorm (37) is an important technique to stabilize the training procedure, we extend to apply it to
 239 SO(2) irreducible representations $\mathbf{x}^{(m)}$, defined as
 240

$$241 \quad \text{LN}(\mathbf{x}^{(m)}) = \frac{\mathbf{x}^{(m)}}{\text{norm}(\mathbf{x}^{(m)})} \circ \left(\frac{\text{norm}(\mathbf{x}^{(m)}) - \mu^{(m)}}{\sigma^{(m)}} \circ g^{(m)} + b^{(m)} \right), \quad (6)$$

242 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
 243 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
 244 norm over the channels. After normalization, learnable affine parameters are introduced with a scale
 245 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
 246 scale of the SO(2) equivariant features with $m > 0$. We refer to it as norm-based Layer Normalization
 247 (LN), consistent with Eq. 6.
 248

249 **SO(2) Tensor Product (TP).** Similar to the tensor product used for SO(3) features, the SO(2) tensor
 250 product provides a mechanism to fuse SO(2) features. Unlike previous operations that treat SO(2)
 251 irreps with each order m separately, the SO(2) tensor product enables interactions between irreps
 252 with different order m . Specifically, the formulation is given by
 253

$$254 \quad \mathbf{z}^{(m_o)} = \mathbf{x}_1^{(m_1),+1} \otimes \mathbf{x}_2^{(m_2),+1} = \begin{pmatrix} \mathbf{x}_{1,-m_1}^{(m_1)} \mathbf{x}_{2,-m_2}^{(m_2)} + \mathbf{x}_{1,-m_1}^{(m_1)} \mathbf{x}_{2,-m_2}^{(m_2)} \\ \mathbf{x}_{1,-m_1}^{(m_1)} \mathbf{x}_{2,-m_2}^{(m_2)} - \mathbf{x}_{1,-m_1}^{(m_1)} \mathbf{x}_{2,-m_2}^{(m_2)} \end{pmatrix}, \quad (7)$$

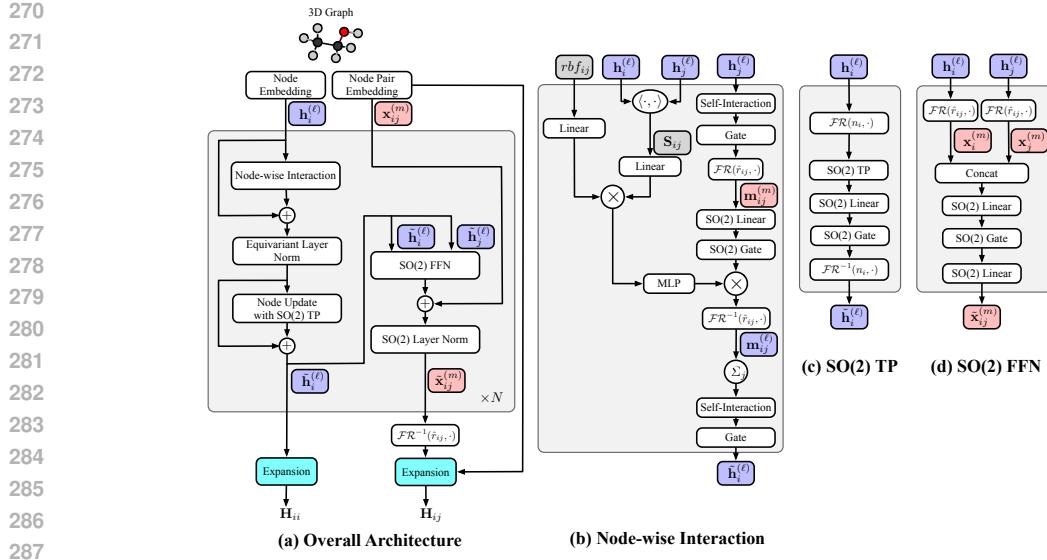
255 where $m_o = m_1 + m_2$, and it is the multiplication between two complex number fusing SO(2)
 256 features from different m .
 257

258 Similarly, when $m_o = m_1 - m_2$ with $m_1 > m_2$, the corresponding SO(2) TP can be formulated as
 259

$$260 \quad \mathbf{z}^{(m_o)} = \mathbf{x}_1^{(m_1),+1} \otimes \mathbf{x}_2^{(m_2),-1} = \begin{pmatrix} \mathbf{x}_{1,-m_1}^{(m_1)} \mathbf{x}_{2,-m_2}^{(m_2)} - \mathbf{x}_{1,-m_1}^{(m_1)} \mathbf{x}_{2,-m_2}^{(m_2)} \\ \mathbf{x}_{1,-m_1}^{(m_1)} \mathbf{x}_{2,-m_2}^{(m_2)} + \mathbf{x}_{1,-m_1}^{(m_1)} \mathbf{x}_{2,-m_2}^{(m_2)} \end{pmatrix} \quad (8)$$

261 Given the input SO(2) irreps with maximum order M_{max} , SO(2) TP includes all valid paths satisfying
 262 $m_o = m_1 + m_2$, $m_o = m_1 - m_2$, $m_1 > m_2$ or $m_o = m_2 - m_1$, $m_2 > m_1$, where $0 \leq m_1, m_2 \leq$
 263 M_{max} . And the number of paths is $O(M_{max}^2)$.
 264

265 Moreover, inspired by the symmetric contraction module proposed in MACE (16) which uses the
 266 generalized CG to fuse multiple SO(3)-equivariant features after aggregating the node features to
 267 perform many-body interactions, a continuous SO(2) TP can be implemented in a similar way to
 268



288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323

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.

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

$$z^{m_o} = \underbrace{\mathbf{x}^{(m_1), s_1} \otimes \mathbf{x}^{(m_2), s_2} \otimes \dots \otimes \mathbf{x}^{(m_v), s_v}}_{(v-1) \text{ TP}}. \quad (9)$$

Considering a set of SO(2) irreducible representations with maximum order M_{max} , the number of paths for SO(2) TP is $O(M_{max}^v)$. For example, the computational cost for simple SO(2) TP between two SO(2) irreps will be $O(M_{max}^2)$, where M_{max} is the cutoff of the order on the SO(2) irreps. In our implementation, we set $M_{max} = L_{max}$, since the SO(2) irreps are transferred from the SO(3) irreps whose cutoff is L_{max} . To see why the time complexity is linear in M_{max} , consider the computation for a single channel. The SO(2) TP enumerates all valid index pairs (m_1, m_2) that satisfy the SO(2) selection rule $m = m_1 + m_2$ and $m = |m_1 - m_2|$. Because both m_1 and m_2 range from 0 to M_{max} , the number of valid combinations is proportional to M_{max}^2 . For each valid path, the computational work is $O(1)$, consisting of a single complex multiplication between two SO(2) feature components. Therefore, the total complexity will be M^2 for two SO(2) irreps as input of SO(2) TP. Note that the time complexity of rotating into the SO(2) local frame and rotating back is $O(L_{max}^3)$. This is because multiplying an irreducible representation of degree ℓ by the corresponding rotation matrix requires ℓ^2 operations. Summing this cost over all degrees $\ell = 0, 1, \dots, L_{max}$ yields a total complexity proportional to $\sum_{\ell=0}^{L_{max}} \ell^2 = O(L_{max}^3)$.

Relationship between SO(2) operations and SO(3) operations. We show the connections between SO(3) TP and SO(2) linear, SO(2) Gate and SO(3) Gate in Appendix D. We use SO(2) TP as a way to fuse SO(2) irreps and demonstrate its help in improving the performance in Table 4. Currently, we regard SO(2) TP and SO(3) TP as non-equivalent operations. While SO(2) TP offers lower computational complexity, it does not necessarily guarantee superior performance compared to SO(3) TP.

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

$$\mathbf{x}_{ij}^{m=0} = \text{MLP}(\text{Linear}(\mathbf{s}_{ij}) \circ \text{Linear}(\text{rbf}(\bar{r}_{ij}))) \quad (10)$$

Node-wise interaction. For the node-wise interaction module, the message passing paradigm of Graph Neural Networks (GNNs) (38) 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 $\text{SO}(3)$ irreps $\mathbf{h}_i^{(\ell)}$. After that, for each message, the $\mathbf{h}_j^{(\ell)}$ is rotated into its local $\text{SO}(2)$ frame $\mathcal{F}(r_{ij})$ obtaining the $\text{SO}(2)$ irreps $\mathbf{m}_{ij}^{(m)}$. Then, $\text{SO}(2)$ Linear and $\text{SO}(2)$ 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

$$\mathbf{s}_{ij} = \left(\langle \mathbf{h}_i^{\ell=0}, \mathbf{h}_j^{\ell=0} \rangle \parallel \langle \mathbf{h}_i^{\ell=1}, \mathbf{h}_j^{\ell=1} \rangle \parallel \cdots \parallel \langle \mathbf{h}_i^{\ell=\ell_{max}}, \mathbf{h}_j^{\ell=\ell_{max}} \rangle \right), \quad (11)$$

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

$$\tilde{\mathbf{m}}_{ij}^{(m)} = \text{SO}(2)\text{Linear}(\mathbf{m}_{ij}^{(m)}) \circ \text{MLP}(\text{Linear}(\mathbf{s}_{ij}) \circ \text{Linear}(\text{rbf}(\bar{r}_{ij}))). \quad (12)$$

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

First, we need to find the reference unit vector for each node to build the local frame on nodes. Specifically, for node n_i , we select the direction vector from the closest neighbor node to the center node to build the local $\text{SO}(2)$ frame, denoted as

$$\mathcal{F}(n_i) = \mathcal{F}(\hat{r}_{ij}) \text{ with } \arg \min_{j \in \mathcal{N}_i} \bar{r}_{ij}, \quad (13)$$

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

$$\mathbf{x}_i^{(m)} = \mathcal{F}\mathcal{R}\left(n_i, \mathbf{h}_i^{(\ell)}\right), \tilde{\mathbf{x}}_i^{(m)} = \text{SO}(2)\text{Linear}\left(\text{SO}(2)\text{TP}\left(\mathbf{x}_i^{(m)}, v\right)\right), \tilde{\mathbf{h}}_i^{(\ell)} = \mathcal{F}\mathcal{R}^{-1}\left(n_i, \tilde{\mathbf{x}}_i^{(m)}\right), \quad (14)$$

where the $\text{SO}(2)$ TP operation collects the $\text{SO}(2)$ irreps according to Eq. 9 with $v - 1$ consecutive $\text{SO}(2)$ TP operations. The collected $\text{SO}(2)$ irreps are then fed into a $\text{SO}(2)$ Linear before being transformed back to the global coordinate system. We show in Table 4 that with the $\text{SO}(2)$ TP within the local frames on nodes can improve the performance.

Although selecting the nearest neighbor provides a fast and simple way to determine the local reference vector for each node, this approach can lead to discontinuities in frame construction (39). Previous work (40) addressed this issue by constructing $O(n^3)$ frames to ensure global continuity. Further improvements can be made by averaging over all local frames derived from the directions between each node and its neighbors, rather than selecting only a single frame per node as in the current framework. This approach requires building $O(n)$ local frames per atom, resulting in $O(n^2)$ frames in total.

Off-diagonal feature updating with $\text{SO}(2)$ 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 $\text{SO}(2)$ local frame and its off-diagonal matrix block. Therefore, we keep all the features within this $\text{SO}(2)$ 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 $\text{SO}(2)$ irreps $\mathbf{m}_i^{(m)}$ and $\mathbf{m}_j^{(m)}$. Hence, the associated off-diagonal feature is given by

$$\mathbf{x}_{ij}^{(m)} = \text{SO}(2)\text{Linear}\left(\text{SO}(2)\text{Gate}\left(\text{SO}(2)\text{Linear}(\mathbf{m}_i^{(m)} \parallel \mathbf{m}_j^{(m)})\right)\right) \quad (15)$$

378 **Overall architecture.** The overall architecture is demonstrated in Figure 2(a). It contains two parts.
 379 The left one updates the node features $\mathbf{h}_i^{(\ell)}$. The skip connection is applied after the node-wise
 380 interaction, and then, the Equivariant LayerNorm (36) is applied before feeding into the $\text{SO}(2)$ TP
 381 operations. The right one is used to update the pairwise features $\mathbf{x}_{ij}^{(m)}$. With the updated pairwise
 382 node features $\tilde{\mathbf{h}}_i$ and $\tilde{\mathbf{h}}_j$, it applies the $\text{SO}(2)$ FFN on them, followed by a skip connection with $\text{SO}(2)$
 383 LayerNorm to update the pairwise $\text{SO}(2)$ irreps $\tilde{\mathbf{x}}_{ij}^{(m)}$.
 384

385 **Matrix Construction.** We follow the expansion module in QHNet (10), which uses the
 386 $(\bar{\otimes} w^{\ell_3})_{(m_1, m_2)}^{(\ell_1, \ell_2)} = \sum_{m_3=-\ell_3}^{\ell_3} C_{(\ell_3, m_3)}^{(\ell_1, m_1), (\ell_2, m_2)} w_{m_3}^{\ell_3}$ and establishes a mapping between the irrep
 387 blocks and the orbital pairs within the Hamiltonian matrix blocks. For the diagonal matrix block,
 388 the node feature $\tilde{\mathbf{h}}^{(\ell)}$ is directly used as the input of Expansion module. For the off-diagonal matrix
 389 blocks, the $\text{SO}(2)$ irreps are first transformed back to the global coordinate system and then fed into
 390 the Expansion module.
 391

393 4 RELATED WORKS

395 Since intrinsic symmetries present in physical systems play an important role in modeling, equivariant
 396 neural networks (41) are explicitly designed to encode these symmetries directly into the architecture.
 397 By construction, these models preserve equivariant features throughout all layers, ensuring that
 398 symmetry-consistent representations are maintained. For Cartesian equivariant models such as
 399 PaiNN (42), ViSNet (43), and GotenNet (41), the architectures rely on scalars and vectors features
 400 within the model. These model are both efficient and powerful, enabling the models to achieve
 401 strong performance across a wide range of tasks. In contrast, spherical equivariant models are built
 402 on $\text{SO}(3)$ irreducible representations and make extensive use of spherical harmonics and tensor
 403 products to rigorously encode rotational symmetries. Examples include TFN (25), SEGNN (23),
 404 SE(3)-Transformers (26), NeQuIP (28), MACE (16), Allegro (27) and Equiformer (29). These models
 405 build their architectures around tensor products as the central mechanism for encoding directional
 406 information within the irreducible representations of the feature space. Although tensor products
 407 offer a powerful mechanism for encoding geometric information, their computational cost has been
 408 widely discussed as a major bottleneck with a computational complexity $O(L^6)$, where L is the
 409 maximum degree of the input irreducible representations. To improve computational efficiency,
 410 ESCN (34) proposes to reduce the $\text{SO}(3)$ tensor product into $\text{SO}(2)$ in this case that the $\text{SO}(3)$ tensor
 411 product can be equivalently replaced by a $\text{SO}(2)$ Linear. This approximation reduces the complexity
 412 to $O(L^3)$, enabling substantially faster training and inference while preserving essential symmetry
 413 properties, as demonstrated in EquiformerV2 (36). Motivated by the efficiency gains achieved through
 414 $\text{SO}(2)$ -based operations, in this work we further generalize the $\text{SO}(2)$ operations beyond $\text{SO}(2)$ -linear
 415 layers. We show that any $\text{SO}(2)$ operations can be incorporated within these $\text{SO}(2)$ local frames
 416 while maintaining global $\text{SO}(3)$ equivariance, enabled by a minimal frame-averaging mechanism. In
 417 addition, these $\text{SO}(2)$ local frames can be constructed not only on edges but also on nodes or other
 418 locations where a well-defined directional reference can be established.

419 Numerous studies have explored the task of Hamiltonian matrix prediction in quantum [systems](#) (11; 44; 45). SchNorb (46) takes the invariant SchNet (18) as its backbone and build models
 420 for directly predicting the Hamiltonian matrix blocks. Subsequent models such as PhiSNet (8) and
 421 QHNet (10) incorporate equivariance into Hamiltonian modeling by leveraging tensor product (TP)
 422 operations and equivariant neural networks to respect the underlying rotational symmetries of the
 423 system. SPHNet (14) introduces sparse gate on tensor product mechanism to accelerate equivariant
 424 computations while maintaining competitive performance. WANet (13) addresses the nonlinearity of
 425 the Hamiltonian’s eigenvalues—*i.e.*, the energy spectrum—by introducing an auxiliary loss function
 426 that directly constrains the predicted eigenvalues, enhancing both accuracy and physical consistency.
 427 For materials datasets, the DeepH series of works (47; 48; 49; 50; 51) framework demonstrates the
 428 evolution from invariant to fully equivariant architectures, capable of predicting both Hamiltonians
 429 and higher-order tensors along material trajectories. HamGNN (12) takes use of TP to build the
 430 equivariant graph networks for its prediction, and DeePTB (15) designs a strictly local model that
 431 efficiently captures many-body interactions. As a plug-in module, TraceGrad (52) introduces an
 432 auxiliary objective based on the norm of the predicted Hamiltonian blocks and corresponding blocks
 433 for improving the performance of Hamiltonian matrix prediction. For the self-consistent training

432
 433 Table 1: Performance on QH9 dataset. The unit for the Hamiltonian \mathbf{H} and eigen energies ϵ is Hartree
 434 denoted by E_h . Lower is better for \mathbf{H} and ϵ ; higher is better for ψ . The best performance scores are
 435 highlighted in **bold**. Underline indicates the second best performance scores.

436 Dataset	437 Model	438 $\mathbf{H} [10^{-6}E_h] \downarrow$	439 off-diagonal	440 all	441 $\epsilon [10^{-6}E_h] \downarrow$	442 $\psi [10^{-2}] \uparrow$
437 QH9-stable-id	QHNet	111.21	73.68	76.31	<u>798.51</u>	95.85
	WANet	—	—	79.99	<u>833.61</u>	96.86
	SPHNet	—	—	<u>45.48</u>	<u>334.28</u>	<u>97.75</u>
	QHNetV2	73.62	28.30	31.50	417.89	98.58
438 QH9-stable-ood	QHNet	111.72	69.88	72.11	<u>644.17</u>	93.68
	SPHNet	—	—	<u>43.33</u>	<u>186.40</u>	98.16
	QHNetV2	61.09	20.81	22.97	165.89	97.68
439 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
440 QH9-dynamic-300k-mol	QHNet	261.63	108.70	119.66	2178.15	90.72
	QHNetV2	138.26	42.06	49.01	629.64	97.43

445 framework (53; 54; 55), it combines the machine models with a self-consistent field (SCF) loop
 446 during training, enabling unsupervised learning of quantum properties. Although this method en-
 447 hances accuracy and expressiveness, it introduces significant computational overhead due to iterative
 448 refinement.

450 5 EXPERIMENTS

451 We evaluate and benchmark our model on QH9 in Section 5.1 and MD17 in Section 5.2. Moreover,
 452 we provide the efficiency studies in Appendix A.1 and ablation studies in Appendix A.2 of SO(2)
 453 TP and SO(2) FFNs, demonstrating the contributions of each proposed component. We notice that a
 454 new dataset, PubChemQH (13), has recently been introduced with larger molecule size and orbital
 455 size. However, as this dataset is not yet publicly available, we will leave the experiments on this
 456 dataset until access becomes open. Our code implementation is based on PyTorch 2.4.1 (56), PyTorch
 457 Geometric 2.6.1 (57), and e3nn (58). In experiments, we train models on 80GB Nvidia A100 with
 458 Intel(R) Xeon(R) Gold 6258R CPU @ 2.70GHz or 46GB NVIDIA RTX 6000 with AMD EPYC
 459 9684X 96-Core Processor for QH9 and 11GB Nvidia GeForce RTX 2080Ti GPU with Intel Xeon
 460 Gold 6248 CPU for MD17.

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

464 5.1 QH9

465 **Datasets.** The QH9 dataset (11) (CC BY-NC SA 4.0 license) consists of four distinct tasks. The
 466 QH9-stable includes 130k molecules derived from QM9 (59; 60). The detailed description of QH9
 467 dataset can be found in Appendix F.1.

468 **Training Details.** Our model follows the same training settings as the QH9 benchmark, with the
 469 corresponding hyperparameters provided in Table 6 in the Appendix F.1.

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

475 5.2 MD17

476 **Datasets.** The MD17 datasets (46) (CC BY-NC license) consists of four molecular trajectories for
 477 water, ethanol, malondialdehyde, and uracil, respectively. The corresponding number of geometries
 478 for each trajectory for train/val/test split is provided in Table 7.

479 **Training Details.** The training hyperparameters are shown in Table 8 following the settings and split
 480 of previous works. **We follow the experimental settings of QHNet on MD17, and the experimental**

486
 487 Table 2: Performance on MD17 dataset. The unit for the Hamiltonian \mathbf{H} and eigen energies ϵ is
 488 Hartree denoted by E_h . Lower is better for \mathbf{H} and ϵ ; higher is better for ψ . The best performance
 489 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)	<u>17.59</u>	<u>85.53</u>	100.00
	QHNet	LSW (10,000, 200,000)	10.36	<u>36.21</u>	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)	<u>21.52</u>	<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

503
 504 results of QHNet and PhiSNet are from Table 1 in QHNet experiments. Since increasing the number
 505 of iterations can improve the final performance, to make a fair comparison, the total number of
 506 iterations are fixed for these experiments with a total 200,000 iterations. The original implementation
 507 and experiments of PhiSNet follows the settings of using reduce on the Plateau and won't stop until
 508 the learning rate is reduced to or achieving a total number of iterations. This will lead to a very large
 509 number of iterations for training, and the number of iterations are not fixed and pretty large for all the
 510 models.

511 **Results.** The results of MD17 is summarized in Table 2. We can find that in ethanol, malondialdehyde
 512 and uracil, our model provides a significant better performance on the MAE for \mathbf{H} with at least 42%
 513 error reduction. Meanwhile, our model provides a better MAE on ϵ . For the water dataset, as shown
 514 in Table 7, it contains only 500 geometries compared to 25,000 geometries for each of the other three
 515 molecules. It indicates that our model tends to achieve better performance when the training set
 516 includes a larger number of geometries.

577 6 CONCLUSION

598 In this work, we propose a novel network QHNetV2 for Hamiltonian matrix prediction that leverages
 599 the $\text{SO}(2)$ local frame to achieve global $\text{SO}(3)$ equivariance while eliminating the need for tensor
 600 product (TP) operations in both diagonal and off-diagonal components. By introducing the $\text{SO}(2)$
 601 local frame, we develop a set of new $\text{SO}(2)$ -equivariant operations to construct powerful and efficient
 602 neural networks. The proposed approach ensures high computational efficiency by avoiding costly
 603 $\text{SO}(3)$ TPs entirely. Experimental results on the QH9 and MD17 datasets demonstrate the effectiveness
 604 of our method. Our work opens the door to further incorporating efficient and powerful $\text{SO}(2)$ -based
 605 operations and frames into geometric deep learning, while preserving global $\text{SO}(3)$ equivariance.

540 REFERENCES

541

542 [1] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136:B864–B871, 1964.

543 [2] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140:A1133–A1138, 1965.

544

545 [3] W. Kohn. Nobel lecture: Electronic structure of matter—wave functions and density functionals. *Reviews of Modern Physics*, 71:1253–1266, 1999.

546

547 [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.

548

549

550

551

552

553

554

555

556 [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, Rishabh Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Bergammer, 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.

557

558

559

560

561

562

563

564 [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.

565

566

567

568

569

570

571 [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.

572

573

574 [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.

575

576

577 [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.

578

579

580 [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.

581

582

583

584 [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.

585

586

587

588 [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.

589

590

591

592 [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.

593

594 [14] Erpai Luo, Xinran Wei, Lin Huang, Yunyang Li, Han Yang, Zun Wang, Chang Liu, Zaishuo
 595 Xia, Jia Zhang, and Bin Shao. Efficient and scalable density functional theory Hamiltonian
 596 prediction through adaptive sparsity. *arXiv preprint arXiv:2502.01171*, 2025.

597
 598 [15] Zhanghao Zhouyin, Zixi Gan, Shishir Kumar Pandey, Linfeng Zhang, and Qiangqiang Gu.
 599 Learning local equivariant representations for quantum operators. In *The Thirteenth Interna-
 600 tional Conference on Learning Representations*, 2025.

601 [16] Ilyes Batatia, David Peter Kovacs, Gregor N. C. Simm, Christoph Ortner, and Gabor Csanyi.
 602 MACE: Higher order equivariant message passing neural networks for fast and accurate force
 603 fields. In *Advances in Neural Information Processing Systems*, 2022.

604
 605 [17] Attila Szabo and Neil S Ostlund. *Modern Quantum Chemistry: Introduction to Advanced
 606 Electronic Structure Theory*. Courier Corporation, 2012.

607
 608 [18] Kristof T Schütt, Huziel E Sauceda, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller.
 609 SchNet – a deep learning architecture for molecules and materials. *The Journal of Chemical
 610 Physics*, 148(24):241722, 2018.

611 [19] Johannes Gasteiger, Janek Groß, and Stephan Günemann. Directional message passing for
 612 molecular graphs. In *International Conference on Learning Representations*, 2020.

613 [20] Yi Liu, Limei Wang, Meng Liu, Yuchao Lin, Xuan Zhang, Bora Oztekin, and Shuiwang Ji.
 614 Spherical message passing for 3D molecular graphs. In *International Conference on Learning
 615 Representations*, 2022.

616 [21] Limei Wang, Yi Liu, Yuchao Lin, Haoran Liu, and Shuiwang Ji. ComENet: Towards complete
 617 and efficient message passing for 3D molecular graphs. In *The 36th Annual Conference on
 618 Neural Information Processing Systems*, pages 650–664, 2022.

619
 620 [22] Guillem Simeon and Gianni De Fabritiis. TensorNet: Cartesian tensor representations for
 621 efficient learning of molecular potentials. *Advances in Neural Information Processing Systems*,
 622 36:37334–37353, 2023.

623
 624 [23] Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J Bekkers, and Max Welling.
 625 Geometric and physical quantities improve $E(3)$ equivariant message passing. In *International
 626 Conference on Learning Representations*, 2022.

627
 628 [24] Yuanqing Wang and John Chodera. Spatial attention kinetic networks with $E(n)$ -equivariance.
 629 In *The Eleventh International Conference on Learning Representations*, 2023.

630
 631 [25] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick
 632 Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3D point
 633 clouds. *arXiv Preprint, arXiv:1802.08219*, 2018.

634
 635 [26] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. SE(3)-transformers: 3D
 636 roto-translation equivariant attention networks. *Advances in Neural Information Processing
 637 Systems*, 33:1970–1981, 2020.

638
 639 [27] Albert Musaelian, Simon Batzner, Anders Johansson, Lixin Sun, Cameron J Owen, Mordechai
 640 Kornbluth, and Boris Kozinsky. Learning local equivariant representations for large-scale
 641 atomistic dynamics. *Nature Communications*, 14(1):579, 2023.

642
 643 [28] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai
 644 Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. $E(3)$ -equivariant graph neu-
 645 ral networks for data-efficient and accurate interatomic potentials. *Nature Communications*,
 646 13(1):2453, 2022.

647
 648 [29] Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant graph attention transformer for 3D
 649 atomistic graphs. In *The Eleventh International Conference on Learning Representations*, 2023.

650
 651 [30] Klaus-Robert Müller, Oliver Thorsten Unke, and Thorben Frank. So3krates - self-attention for
 652 higher-order geometric interactions on arbitrary length-scales. 2022.

648 [31] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural
649 networks. In *International Conference on Machine Learning*, pages 9323–9332. PMLR, 2021.
650

651 [32] Shengjie Luo, Tianlang Chen, and Aditi S Krishnapriyan. Enabling efficient equivariant
652 operations in the Fourier basis via Gaunt Tensor Products. In *The Twelfth International
653 Conference on Learning Representations*.

654 [33] Zhao Xu, Haiyang Yu, Montgomery Bohde, and Shuiwang Ji. Equivariant graph network
655 approximations of high-degree polynomials for force field prediction. *Transactions on Machine
656 Learning Research*, 2024.

657 [34] Saro Passaro and C Lawrence Zitnick. Reducing SO(3) convolutions to SO(2) for efficient
658 equivariant gnns. In *International Conference on Machine Learning*, pages 27420–27438.
659 PMLR, 2023.

660 [35] Yuchao Lin, Jacob Helwig, Shurui Gui, and Shuiwang Ji. Equivariance via minimal frame aver-
661 aging for more symmetries and efficiency. In *Proceedings of the 41st International Conference
662 on Machine Learning*, pages 30042–30079, 2024.

664 [36] Yi-Lun Liao, Brandon M Wood, Abhishek Das, and Tess Smidt. EquiformerV2: Improved
665 equivariant transformer for scaling to higher-degree representations. In *The Twelfth International
666 Conference on Learning Representations*, 2024.

667 [37] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint
668 arXiv:1607.06450*, 2016.

670 [38] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural
671 message passing for quantum chemistry. In *International conference on machine learning*,
672 pages 1263–1272. PMLR, 2017.

673 [39] Nadav Dym, Hannah Lawrence, and Jonathan W Siegel. Equivariant frames and the impossibil-
674 ity of continuous canonicalization. *arXiv preprint arXiv:2402.16077*, 2024.

676 [40] Sergey Pozdnyakov and Michele Ceriotti. Smooth, exact rotational symmetrization for deep
677 learning on point clouds. *Advances in Neural Information Processing Systems*, 36:79469–79501,
678 2023.

679 [41] Alexandre Duval, Simon V Mathis, Chaitanya K Joshi, Victor Schmidt, Santiago Miret,
680 Fragkiskos D Malliaros, Taco Cohen, Pietro Lio, Yoshua Bengio, and Michael Bronstein.
681 A hitchhiker’s guide to geometric gnns for 3d atomic systems. *arXiv preprint arXiv:2312.07511*,
682 2023.

683 [42] Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the
684 prediction of tensorial properties and molecular spectra. In *International Conference on
685 Machine Learning*, pages 9377–9388. PMLR, 2021.

687 [43] Yusong Wang, Shaoning Li, Xinheng He, Mingyu Li, Zun Wang, Nanning Zheng, Bin Shao,
688 Tie-Yan Liu, and Tong Wang. Visnet: an equivariant geometry-enhanced graph neural network
689 with vector-scalar interactive message passing for molecules. *arXiv preprint arXiv:2210.16518*,
690 2022.

691 [44] Kuzma Khrabrov, Anton Ber, Artem Tsypin, Konstantin Ushenin, Egor Rumiantsev, Alexander
692 Telepov, Dmitry Protasov, Ilya Shenbin, Anton Alekseev, Mikhail Shirokikh, et al. ∇^2 DFT: A
693 universal quantum chemistry dataset of drug-like molecules and a benchmark for neural network
694 potentials. *Advances in Neural Information Processing Systems*, 37:36869–36889, 2024.

695 [45] Kuzma Khrabrov, Ilya Shenbin, Alexander Ryabov, Artem Tsypin, Alexander Telepov, Anton
696 Alekseev, Alexander Grishin, Pavel Strashnov, Petr Zhilyaev, Sergey Nikolenko, et al. nablaDFT:
697 Large-scale conformational energy and Hamiltonian prediction benchmark and dataset. *Physical
698 Chemistry Chemical Physics*, 24(42):25853–25863, 2022.

700 [46] Kristof T Schütt, Michael Gastegger, Alexandre Tkatchenko, K-R Müller, and Reinhard J
701 Maurer. Unifying machine learning and quantum chemistry with a deep neural network for
molecular wavefunctions. *Nature Communications*, 10(1):5024, 2019.

702 [47] He Li, Zun Wang, Nianlong Zou, Meng Ye, Runzhang Xu, Xiaoxun Gong, Wenhui Duan,
 703 and Yong Xu. Deep-learning density functional theory Hamiltonian for efficient *ab initio*
 704 electronic-structure calculation. *Nature Computational Science*, 2(6):367–377, 2022.

705 [48] Xiaoxun Gong, He Li, Nianlong Zou, Runzhang Xu, Wenhui Duan, and Yong Xu. General
 706 framework for E(3)-equivariant neural network representation of density functional theory
 707 Hamiltonian. *Nature Communications*, 14(1):2848, 2023.

708 [49] Yuxiang Wang, He Li, Zechen Tang, Honggeng Tao, Yanzhen Wang, Zilong Yuan, Zezhou
 709 Chen, Wenhui Duan, and Yong Xu. DeepH-2: enhancing deep-learning electronic structure via
 710 an equivariant local-coordinate transformer. *arXiv preprint arXiv:2401.17015*, 2024.

711 [50] He Li, Zechen Tang, Xiaoxun Gong, Nianlong Zou, Wenhui Duan, and Yong Xu. Deep-learning
 712 electronic-structure calculation of magnetic superstructures. *Nature Computational Science*,
 713 3(4):321–327, 2023.

714 [51] Zilong Yuan, Zechen Tang, Honggeng Tao, Xiaoxun Gong, Zezhou Chen, Yuxiang Wang, He Li,
 715 Yang Li, Zhiming Xu, Minghui Sun, et al. Deep learning density functional theory Hamiltonian
 716 in real space. *arXiv preprint arXiv:2407.14379*, 2024.

717 [52] Shi Yin, Xinyang Pan, Fengyan Wang, and Lixin He. TraceGrad: a framework learning
 718 expressive SO(3)-equivariant non-linear representations for electronic-structure Hamiltonian
 719 prediction. *arXiv preprint arXiv:2405.05722*, 2024.

720 [53] Feitong Song and Ji Feng. NeuralSCF: Neural network self-consistent fields for density
 721 functional theory. *arXiv preprint arXiv:2406.15873*, 2024.

722 [54] He Zhang, Chang Liu, Zun Wang, Xinran Wei, Siyuan Liu, Nanning Zheng, Bin Shao, and
 723 Tie-Yan Liu. Self-consistency training for density-functional-theory Hamiltonian prediction. In
 724 *International Conference on Machine Learning*, pages 59329–59357. PMLR, 2024.

725 [55] Alexander Mathiasen, Hatem Helal, Paul Balanca, Adam Krzywaniak, Ali Parviz, Frederik
 726 Hvilshøj, Blazej Banaszewski, Carlo Luschi, and Andrew William Fitzgibbon. Reducing the
 727 cost of quantum chemical data by backpropagating through density functional theory. *arXiv
 728 preprint arXiv:2402.04030*, 2024.

729 [56] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan,
 730 Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative
 731 style, high-performance deep learning library. *Advances in Neural Information Processing
 732 Systems*, 32, 2019.

733 [57] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with PyTorch Geometric.
 734 *arXiv Preprint, arXiv:1903.02428*, 2019.

735 [58] Mario Geiger, Tess Smidt, Alby M., Benjamin Kurt Miller, Wouter Boomsma, Bradley Dice,
 736 Kostiantyn Lapchевskyi, Maurice Weiler, Michał Tyszkiewicz, Simon Batzner, Dylan Madisetti,
 737 Martin Uhrin, Jes Frellsen, Nuri Jung, Sophia Sanborn, Mingjian Wen, Josh Rackers, Marcel
 738 Rød, and Michael Bailey. Euclidean neural networks: e3nn, 2022.

739 [59] Lars Ruddigkeit, Ruud Van Deursen, Lorenz C Blum, and Jean-Louis Reymond. Enumeration
 740 of 166 billion organic small molecules in the chemical universe database GDB-17. *Journal of
 741 Chemical Information and Modeling*, 52(11):2864–2875, 2012.

742 [60] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quan-
 743 tum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1(1):140022,
 744 2014.

745 [61] Omri Puny, Matan Atzmon, Heli Ben-Hamu, Ishan Misra, Aditya Grover, Edward J Smith, and
 746 Yaron Lipman. Frame averaging for invariant and equivariant network design. *arXiv preprint
 747 arXiv:2110.03336*, 2021.

748 [62] Michael M. Hirsch, Michael S. Gordon, and Michael S. Gordon. A geometric framework for
 749 equivariant neural networks. *arXiv preprint arXiv:2206.04710*, 2022.

750 [63] Michael M. Hirsch, Michael S. Gordon, and Michael S. Gordon. A geometric framework for
 751 equivariant neural networks. *arXiv preprint arXiv:2206.04710*, 2022.

752 [64] Michael M. Hirsch, Michael S. Gordon, and Michael S. Gordon. A geometric framework for
 753 equivariant neural networks. *arXiv preprint arXiv:2206.04710*, 2022.

754 [65] Michael M. Hirsch, Michael S. Gordon, and Michael S. Gordon. A geometric framework for
 755 equivariant neural networks. *arXiv preprint arXiv:2206.04710*, 2022.

756

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

758

759

760

761

762

763

764

765

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.

766

767

768

769

770

771

772

A EXPERIMENTS

773

774

A.1 EFFICIENCY STUDIES

775

776

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.

777

778

779

780

781

782

783

784

A.2 ABLATION STUDIES

785

786

787

788

789

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. **Note that we compare the experimental results of adding these components and simply removing these components (the blocks are skipped).** 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.

790

791

B BACKGROUND

792

793

B.1 DENSITY FUNCTIONAL THEORY

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

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

810 $\Psi(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_{N_e})$. Kohn and Sham proposed a practical approach to map a many-body interacting
 811 system onto a set of non-interacting one-body systems where each electron moves in an effective
 812 potential arising from the nuclei and the average effect of other electrons (2). This leads to the
 813 well-known single-particle Kohn-Sham equation: $\hat{H}_{\text{KS}}|\psi_p\rangle = \epsilon_p|\psi_p\rangle$ for individual electrons, where
 814 \hat{H}_{KS} is the Kohn-Sham Hamiltonian, ψ_p is the single-electron eigen wavefunction, and ϵ_p is the
 815 corresponding eigen energy. The Kohn-Sham Hamiltonian is given by $\hat{H}_{\text{KS}} = \hat{T} + \hat{V}_{\text{Hartree}} + \hat{V}_{\text{XC}} +$
 816 \hat{V}_{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
 817 exchange-correlation potential, and \hat{V}_{ext} is external potential. The electron density from the Kohn-
 818 Sham equation with the exact exchange-correlation energy functional, should be same as that of the
 819 many-body interacting system: $\rho(\mathbf{r}) = \sum_p f_p |\psi_p(\mathbf{r})|^2 d\mathbf{r} = \int |\Psi(\mathbf{r}, \mathbf{r}_2, \dots, \mathbf{r}_{N_e})|^2 d\mathbf{r}_2 \dots d\mathbf{r}_{N_e}$,
 820 where f_p is the occupation number of the p -th Kohn-Sham eigen state. Therefore, solving the full
 821 many-electron wavefunction Ψ is no longer required.
 822

823 B.2 ATOMIC ORBITALS AND HAMILTONIAN MATRICES

825 The Kohn-Sham equation, as the central equation in DFT, can be solved in a predefined basis set,
 826 such as the STO-3G atomic orbital basis which combines radial functions with spherical harmonics
 827 centered on each atom. The calculated Kohn-Sham eigen wavefunction ψ_n , often referred to as
 828 molecular orbital in the context of molecules, can be expressed as a linear combination of these
 829 predefined basis functions $\psi_n(\mathbf{r}) = \sum_p \mathbf{C}_{pn} \phi_p(\mathbf{r})$, known as the Linear Combination of Atomic
 830 Orbital (LCAO) method (17). Each single-electron wavefunction $\psi_n(\mathbf{r})$ depends only on three
 831 spatial variables, thereby avoiding the exponential scaling associated with the $3N_e$ -dimensional full
 832 many-electron wavefunction. By applying the predefined orbitals and the LCAO method within DFT,
 833 the Kohn-Sham equation can be converted into the following matrix form:
 834

$$835 \mathbf{H}\mathbf{C} = \epsilon\mathbf{S}\mathbf{C}, \quad (16)$$

836 where \mathbf{H} is the Hamiltonian matrix of size $\mathbb{R}^{N_o \times N_o}$. Each matrix element is defined by evaluating
 837 the Hamiltonian operator in a pair of predefined atomic orbitals, shown as $\mathbf{H}_{pq} \equiv \langle \phi_p | \hat{H}_{\text{KS}} | \phi_q \rangle =$
 838 $\int \phi_p^*(\mathbf{r}) \hat{H}_{\text{KS}} \phi_q(\mathbf{r}) d\mathbf{r}$. Here, N_o denotes the number of predefined orbitals which typically increases
 839 linearly with the number of electrons N_e . $\mathbf{C} \in \mathbb{R}^{N_o \times N_o}$ denotes the wavefunction coefficient
 840 matrix. $\epsilon \in \mathbb{R}^{N_o \times N_o}$ is a diagonal matrix where each diagonal element corresponds to the eigen
 841 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
 842 of a pair of predefined basis over the spatial space $\mathbf{S}_{pq} = \int \phi_p^*(\mathbf{r}) \phi_q(\mathbf{r}) d\mathbf{r}$.
 843

844 C SO(3) EQUIVARIANCE OF SO(2) LOCAL FRAMES

846 Frame averaging (61; 39; 35) is a technique that enforces equivariance to any model Φ by an
 847 $\text{SO}(3)$ -equivariant frame $\mathcal{F} : \mathbb{R}^3 \rightarrow \text{SO}(3)$ such that
 848

$$849 \langle \Phi \rangle_{\mathcal{F}}(\hat{r}) = \frac{1}{|\mathcal{F}(\hat{r})|} \sum_{g \in \mathcal{F}(\hat{r})} g \cdot \Phi(g^{-1} \cdot \hat{r}) \quad (17)$$

852 With a little abuse of the definition, we further consider the input space includes the $\text{SO}(3)$ irreps
 853 which is build upon the global coordinate system. Since there are many candidate frame constructions
 854 for frame averaging, we use the minimal frame averaging (35) to give an analytic frame construction
 855 with a minimal frame size. To apply the minimal frame averaging (35), the canonicalization is
 856 required to transfer the input features into its canonical form, defined in Definition 3.3 (35). We
 857 define the canonicalization operation based on a reference vector \hat{r} as $c(\hat{r}) = h^{-1} \cdot \hat{r} = \hat{v}$, where a
 858 rotation $h \in \text{SO}(3)$ is applied to align the reference vector \hat{r} with the fixed target vector \hat{v} .
 859

860 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
 861

$$862 \text{Stab}_{\text{SO}(3)}(\hat{v}) = \{g \in \text{SO}(3) \mid g \cdot \hat{v} = \hat{v}\}.$$

863 Instead of employing an arbitrary local model which could make the direct sum over all $g \in$
 864 $\text{Stab}_{\text{SO}(3)}(\hat{v})$ computationally expensive or even intractable, since $\text{Stab}_{\text{SO}(3)}(\hat{v})$ is isomorphic to

864 SO(2) (rotations about the y -axis), we consider Φ to be $\text{Stab}_{\text{SO}(3)}(\hat{v})$ -equivariant with respect to
 865 the reference axis r . That's to say, for any rotation $g \in \text{Stab}_{\text{SO}(3)}(\hat{v})$, the local network satisfies
 866 $\Phi(g \cdot \hat{r}) = g\Phi(\hat{r})$. In this case, the frame-averaging sum collapses such that
 867

$$\begin{aligned}
 868 \langle \Phi \rangle_{\mathcal{F}}(\hat{r}) &= \frac{1}{|\mathcal{F}(\hat{r})|} \sum_{g \in \mathcal{F}(\hat{r})} g \cdot \Phi(g^{-1}\hat{r}) \\
 869 &= \frac{1}{|\text{Stab}_{\text{SO}(3)}(\hat{v})|} \sum_{g \in \text{Stab}_{\text{SO}(3)}(\hat{v})} hg \cdot \Phi((hg)^{-1} \cdot \hat{r}) \\
 870 &= \frac{1}{|\text{Stab}_{\text{SO}(3)}(\hat{v})|} \sum_{g \in \text{Stab}_{\text{SO}(3)}(\hat{v})} hg \cdot \Phi(g^{-1}h^{-1} \cdot \hat{r}) \\
 871 &= \frac{1}{|\text{Stab}_{\text{SO}(3)}(\hat{v})|} \sum_{g \in \text{Stab}_{\text{SO}(3)}(\hat{v})} hgg^{-1} \cdot \Phi(h^{-1} \cdot \hat{r}) \\
 872 &= \frac{1}{|\text{Stab}_{\text{SO}(3)}(\hat{v})|} \sum_{g \in \text{Stab}_{\text{SO}(3)}(\hat{v})} h \cdot \Phi(h^{-1} \cdot \hat{r}) \\
 873 &= h \cdot \Phi(h^{-1} \cdot \hat{r}).
 \end{aligned} \tag{18}$$

883 Thus, the entire frame reduces to the single rotation h making the computation simple and tractable.
 884 We then define the \mathcal{F} built upon the $\text{Stab}_{\text{SO}(3)}(r)$ -equivariant Φ as the *SO(2) local frame*.
 885

886 D SO(2) OPERATIONS AND SO(3) OPERATIONS

887 D.1 SO(2) LINEAR AND SO(3) TP WITH SPHERICAL HARMONICS

890 The relationship between SO(2) linear operations defined with a local frame and the SO(3) tensor
 891 product between irreducible representations and spherical harmonics has been investigated in the
 892 eSCN paper (34). For completeness, we briefly rehearse their proof here. If so, below we first provide
 893 the specific set of parameters for the SO(2) linear operation with SO(2) local frames that can exactly
 894 reproduce the results of the SO(3) TP with transformed CG coefficients. We will then explain why
 895 SO(2) linear operation with SO(2) local frames has the capacity to replicate it. Note that it is not
 896 to say that SO(2) linear with SO(2) local frame is definitely better, but it should have the potential
 897 to be at least the same. When performing SO(3) tensor product between the rotated SO(3) irreps
 898 and rotated spherical harmonics, the rotated spherical harmonics will be zero in the position with
 899 $m \neq 0$, and 1 for the position $m = 0$. Therefore, we can treat it as the corresponding transformed
 900 CG coefficients are nonzero for the following position.

$$(\mathbf{c}_{l_i, l_f, l_o})_m = \begin{cases} \mathbf{C}_{(l_i, m), (l_f, 0)}^{(l_o, m)} & \text{if } m > 0 \\ \mathbf{C}_{(l_i, 0), (l_f, 0)}^{(l_o, 0)} & \text{if } m = 0 \\ \mathbf{C}_{(l_i, m), (l_f, 0)}^{(l_o, -m)} & \text{if } m < 0 \end{cases} \tag{19}$$

901 In this way, the SO(3) TP can be represented as

$$\mathbf{f}_{m_o}^{(l_o)} = \sum_{m_i} \left(\mathbf{x}_s^{(l_i)} \right)_{m_i} \mathbf{C}_{(l_i, m_i), (l_f, 0)}^{(l_o, m_o)} \mathbf{h}_{l_i, l_f, l_o}, \tag{20}$$

902 where \mathbf{f} is the output irreps and \mathbf{h} is the coefficients for each path.

903 From the proposition in 3.1 from eSCN (34), we have

$$\mathbf{C}_{(l_i, m), (l_f, 0)}^{(l_o, m)} = \mathbf{C}_{(l_i, -m), (l_f, 0)}^{(l_o, -m)}, \mathbf{C}_{(l_i, -m), (l_f, 0)}^{(l_o, m)} = -\mathbf{C}_{(l_i, m), (l_f, 0)}^{(l_o, -m)},$$

904 and $m_o = m_i$ or $m_o = -m_i$.

905 By using this proposition, the previous equation can be deducted into

$$\mathbf{f}_{m_o}^{(l_o)} = \sum_{m_i=m_o, -m_o} \left(\mathbf{x}_s^{(l_i)} \right)_{m_i} \mathbf{C}_{(l_i, m_i), (l_f, 0)}^{(l_o, m_o)} \mathbf{h}_{l_i, l_f, l_o}, \tag{21}$$

918
919
920 And then, if we write $\mathbf{f}_{m_o}^{(l_o)}$ and $\mathbf{f}_{-m_o}^{(l_o)}$ together, it will become
921
922
923
924
925

$$\begin{pmatrix} f_{m_o}^{(l_o)} \\ f_{-m_o}^{(l_o)} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{h}}_{m_o}^{(l',l)} & -\tilde{\mathbf{h}}_{-m_o}^{(l',l)} \\ \tilde{\mathbf{h}}_{-m_o}^{(l',l)} & \tilde{\mathbf{h}}_{m_o}^{(l',l)} \end{pmatrix} \cdot \begin{pmatrix} \left(\mathbf{x}_s^{(l_i)} \right)_{m_o} \\ \left(\mathbf{x}_s^{(l_i)} \right)_{-m_o} \end{pmatrix}, \quad (22)$$

926 where $w_m^{(l_i, l_o)} = \sum_{l_f} h_{l_i, l_f, l_o} \cdot (c_{l_i, l_f, l_o})_m$ is the weight in the SO(2) linear case for $m_o \neq 0$.
927
928 Furthermore, since the parameters in SO(2) linear are learnable, the model may discover another set
929 of parameterizations with lower training loss, potentially achieving better performance. Conceptually,
930 SO(3) TP with transformed CG corresponds to a particular point in the parameter space of the SO(2)
931 linear with SO(2) local frame. If the optimal parameters can be identified within this space, the model
932 should perform at least as well as with SO(3) TP.
933
934

D.2 SO(2) GATE AND SO(3) GATE

935 **SO(3) Gate.** The SO(3) Gate takes the features with $l = 0$ as the input of an MLP, and then provides
936 a gate value that is multiplied on each SO(3) feature with $l > 0$ as well as the new SO(3) features
937 with $l = 0$. For convenience, we assume that the local frame, designed to rotate consistently with the
938 input molecule, is currently aligned with the y -axis direction, so that the rotation procedure can be
939 omitted.
940

941 **SO(2) Gate.** Under this setup, the SO(2) Gate uses $m = 0$ features as the input of its MLP, including
942 both $l = 0$ features and $m = 0, l \neq 0$ features. The MLP then produces a separate gate value for each
943 SO(2) feature with different m , as well as new $m = 0$ features.
944

945 **Comparison.** Because the MLP in the SO(2) Gate includes all the inputs of the SO(3) Gate (i.e.,
946 $l = 0$ components), it can replicate the same behavior as the SO(3) Gate. However, since the MLP
947 contains nonlinear functions and can produce nonlinear results for the $m = 0, l \neq 0$ parts, the SO(3)
948 Gate cannot approximate the SO(2) Gate. By including additional higher-order $m = 0$ inputs and
949 using nonlinear functions to produce $m = 0$ outputs, the SO(2) Gate can therefore learn mappings
950 that the SO(3) Gate cannot express. The details are demonstrated in Table 5.
951
952

Table 5: Comparison between SO(2) Gate and SO(3) Gate.

Aspect	SO(2) Gate	SO(3) Gate
MLP Input	$m = 0$ features (includes $l = 0$ and $m = 0, l > 0$)	$l = 0$ features only
Gate Applied To	Each SO(2) feature with different $m > 0$	Each SO(3) feature with different $l > 0$
New Output Components	$m = 0$ (including both $l = 0$ and $l > 0$)	$l = 0$ only

E EQUIVARIANCE OF THE IRREP MAPPING

963 For the rotation within the local frame, an SO(2) rotation with angle α applied to features of order m
964 corresponds to the following rotation matrix:
965
966

$$R_m(\alpha) = \begin{pmatrix} \cos(m\alpha) & -\sin(m\alpha) \\ \sin(m\alpha) & \cos(m\alpha) \end{pmatrix}. \quad (23)$$

967 For SO(3) rotations acting on SO(3) irreps, if the rotation is restricted to the z -axis (i.e., the exact
968 $m = 0$ feature for $l = 1$), it corresponds to the ZYZ Euler rotation with parameters $(\alpha, \beta, \gamma) =$
969 $(\alpha, 0, 0)$. The corresponding rotation matrix is
970
971

$$D_{mn}^{(\ell)}(\alpha, 0, 0) = e^{im\alpha} \delta_{mn}, \quad (24)$$

972 for $\text{SO}(3)$ irreps with angular momentum ℓ .
 973

974 With the change of basis U_ℓ derived in Equation (25) of the eSCN, the rotation matrix can be expressed
 975 in the same block-diagonal format as

$$976 \quad D_{\text{real}}^{(\ell)}(\alpha) = U_\ell D_{\text{complex}}^{(\ell)}(\alpha, 0, 0) U_\ell^\top = \text{diag}(1, R_1(\alpha), R_2(\alpha), \dots, R_\ell(\alpha)), \quad (25)$$

978 where each block $R_m(\alpha)$ matches the $\text{SO}(2)$ rotation matrix defined above.
 979

980 Then, for the equivariance of the mapping operation $FR(\hat{r})$, the formal definition of equivariance for
 981 a map f is

$$982 \quad f(g \cdot x) = \rho(g) f(x), \quad (26)$$

983 where $g \in \text{SO}(3)$ acts on the input, and $\rho(g)$ is the induced $\text{SO}(2)$ action on the output irreps.
 984

985 In our setting, f denotes the operation mapping from $\text{SO}(3)$ to $\text{SO}(2)$ with the local frame defined
 986 as $FR(r_{ij})$. When $g \in \text{SO}(3)$ is applied to the input space, the corresponding stabilizer rotation
 987 is given by $g'g^{-1}$, where g' is a rotation around the z -axis that maps the rotated direction vector
 988 gr_{ij} back to the z -axis. Applying $g'g^{-1}$ to the rotated input gx is equivalent to applying g' to the
 989 original $\text{SO}(3)$ irreps x before rotation. As shown earlier, the rotation matrix for $\text{SO}(3)$ features
 990 with only $\alpha \neq 0$ reduces to the $\text{SO}(2)$ rotation form. Hence, the stabilizer rotation g' serves as the
 991 corresponding $\rho(g)$ acting on the output $\text{SO}(2)$ irreps.
 992

F HYPERPARAMETER SETTINGS AND MODEL ARCHITECTURES

F.1 QH9

995 In the QH9-stable-id setting, the training, validation and test sets are randomly split. In contrast, the
 996 QH9-stable-ood settings use a split based on molecular size, with validation and test sets consisting
 997 of molecules larger than those in the training set. The QH9-dynamic-300k dataset is an extended
 998 version of the QH9-dynamic benchmark, comprising 300k molecular geometries generated from 3k
 999 molecules, each with a molecular dynamics trajectory of 100 steps. In the QH9-dynamic-300k-geo
 1000 setting, the training, validation, and test sets consist of the same molecules but different geometries
 1001 sampled from their respective trajectories. In contrast, the QH9-dynamic-300k-mol setting is split
 1002 based on different molecules, with all geometries from a given trajectory assigned to the same split.
 1003 This setup is designed to evaluate the model’s ability to generalize to unseen molecules with diverse
 1004 geometries. We follow the model and training hyperparameters in Table 6 to perform our experiments.
 1005

1006 **Table 6: Hyperparameters in QH9.**

1008 Hyperparameters	1009 Values/search space
1009 Batch size	32
1010 Cutoff distance (Bohr)	15
1011 Initial learning rate	5e-4
1012 Final learning rate	1e-7
1013 Learning rate strategy	linear schedule with warmup
1014 Learning rate warmup batches	1,000
1015 # of batches in training	26,000
1016 # of layers	3
1017 L_{\max}	4
1018 hidden irreps	$256 \times 0e+128 \times 1e+64 \times 2e+32 \times 3e+16 \times 4e$
1019 hidden mirreps	$1024 \times 0m+256 \times 1m+64 \times 2m+32 \times 3m+16 \times 4m$
1020 hidden ffn mirreps	$2048 \times 0m+512 \times 1m+256 \times 2m+64 \times 3m+32 \times 4m$

1026 F.2 MD17
1027

1028 The statistics of MD17 is shown in Table 7. We follow the model and training hyperparameters
 1029 in Table 8 to perform our experiments. Specifically, for the model molecule, we choose a
 1030 smaller model since the number of training geometries is smaller. We choose hidden irreps to be
 1031 $64x0e+32x1e+16x2e+8x3e+8x4e$, hidden mirreps to be $64x0m+32x1m+16x2m+8x3m+8x4m$, and
 1032 hidden ffn mirreps to be $64x0m+32x1m+16x2m+8x3m+8x4m$.

1033

1034 Table 7: The statistics of MD17 dataset.
1035

Dataset	Molecule	# of structures	Train	Val	Test
Water	H_2O	4,900	500	500	3,900
Ethanol	C_2H_5OH	30,000	25,000	500	4,500
Malondialdehyde	$CH_2(CHO)_2$	26,978	25,000	500	1,478
Uracil	$C_4H_4N_2O_2$	30,000	25,000	500	4,500

1041

1042

1043 Table 8: Hyperparameters in MD17.
1044

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	$2048x0m+512x1m+256x2m+64x3m+32x4m$

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079