SE(3)-equivariant prediction of molecular wavefunctions and electronic densities

Anonymous Author(s)
Affiliation
Address
email

Abstract

Machine learning has enabled the prediction of quantum chemical properties with high accuracy and efficiency, allowing to bypass computationally costly *ab initio* calculations. Instead of training on a fixed set of properties, more recent approaches attempt to learn the electronic wavefunction (or density) as a central quantity of atomistic systems, from which all other observables can be derived. This is complicated by the fact that wavefunctions transform non-trivially under molecular rotations, which makes them a challenging prediction target. To solve this issue, we introduce general SE(3)-equivariant operations and building blocks for constructing deep learning architectures for geometric point cloud data and apply them to reconstruct wavefunctions of atomistic systems with unprecedented accuracy. Our model reduces prediction errors by up to *two orders of magnitude* (!) compared to the previous state-of-the-art and makes it possible to derive properties such as energies and forces directly from the wavefunction in an end-to-end manner. We demonstrate the potential of our approach in a transfer learning application, where a model trained on low accuracy reference wavefunctions implicitly learns to correct for electronic many-body interactions from observables computed at a higher level of theory. Such machine-learned wavefunction surrogates pave the way towards novel semi-empirical methods, offering resolution at an electronic level while drastically decreasing computational cost. While we focus on physics applications in this contribution, the proposed equivariant framework for deep learning on point clouds is promising also beyond, say, in computer vision or graphics.

1 Introduction

Machine learning methods are becoming increasingly popular in quantum chemistry as a means to circumvent expensive *ab initio* calculations and led to advances in a broad range of applications, including the construction of potential energy surfaces [1–8], prediction of electron densities and density functionals [9–14], and development of models capable of predicting a range of physical observables across chemical space [15–27]. Typically, such models are trained on reference data for a predetermined set of quantum chemical properties and need to be retrained if other properties are required. However, if a model is capable of predicting the wavefunction, expectation values for *any* observable can be derived from it. Unfortunately, such an approach is complicated by the fact that wavefunctions are typically expressed in terms of rotationally equivariant basis functions, introducing non-trivial transformations under molecular rotations, which are difficult to learn from data. To solve this issue, we propose several SE(3)-equivariant operations for deep learning architectures for geometric point cloud data, which capture the effects of rotations without needing to learn them explicitly. We assemble these building blocks to construct PhiSNet, a novel deep learning (DL) architecture for predicting wavefunctions and electronic densities, which is significantly more

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

accurate than non-equivariant models. For the first time, sufficient accuracy is reached to predict properties like energies and forces *directly* from the wavefunction and in *end-to-end manner*.

This makes it possible to learn wavefunctions that lead to modified properties, which is interesting from an inverse design perspective; or the development of novel machine-learned semi-empirical methods, for example by learning a correction to the wavefunction that mimics the effects of electron correlation. Such hybrid methods maintain the accuracy and generality of high level electronic structure calculations while drastically reducing their computational cost.
Beyond physics, other applications of this novel equivariant DL architecture to e.g. computer vision or graphics are conceivable — whenever accurate invariant analyses of high dimensional point clouds are of importance.

In summary, this work provides the following contributions:

- We describe general SE(3)-equivariant operations and building blocks for constructing DL architectures for geometric point cloud data.
- We propose PhiSNet, a neural network for predicting wavefunctions and electronic densities from equivariant atomic representations, ensuring physically correct transformation under rotations.
- We apply PhiSNet to predict wavefunctions and electronic densities of a range of small to medium-sized molecules and show that our model reduces prediction errors of electronic structure properties by a factor of up to two orders of magnitude compared to the previous state-of-the-art.
- We showcase a novel transfer-learning application, where a model trained on low accuracy wavefunctions is adapted to predict properties computed at a higher level of theory by learning a correction that implicitly captures the effects of many-body electron correlation.

2 Related work

Only a small number of studies apply machine learning to the challenging problem of modeling the wavefunction directly [28]. This is usually done by predicting Hamiltonian matrices, from which the wavefunction can be obtained by solving a generalized eigenvalue problem. The earliest such study we are aware of is by Hegde and Bowen [29], who used kernel ridge regression to learn the Hamiltonian matrix for two simple case studies. Later, Schütt et al. [30] proposed the SchNOrb neural network architecture, which constructs the Hamiltonian matrix of molecules in a block-wise manner from atom-pair features. Recently, Li et al. [31] presented a deep neural network architecture for predicting the Hamiltonian matrix of simple periodic crystals. However, none of these models form their predictions in a rotationally equivariant manner, i.e. they need to learn how to predict the Hamiltonian matrix for all possible orientations of the system of interest, which requires large amounts of training data. Even when data augmentation via random rotations [32] or special Hamiltonian representations [33] are used to mitigate this issue, the final model is only approximately equivariant, i.e. properties derived from the wavefunction can change unphysically when the system is rotated or the frame of reference is changed. Here, we draw upon insights from a range of SE(3)-equivariant models [34–44] to ensure that predictions exactly preserve the physically correct dependence with respect to the orientation of inputs (for a more detailed discussion see the supplement).

3 Background

The aim of most quantum chemistry methods is to solve the electronic Schrödinger equation

$$\hat{H}_{\mathrm{el}}\Psi_{\mathrm{el}} = E_{\mathrm{el}}\Psi_{\mathrm{el}}, \tag{1}$$

where $\hat{H}_{\mathrm{el}}$ is the Hamiltonian operator describing the interactions and motion of the electrons, Ψ_{el} is the electronic wavefunction and E_{el} is the ground state energy. After Ψ_{el} is determined, all physical observables (beyond E_{el}) can be derived by applying other operators (similar to $\hat{H}_{\mathrm{el}}$) to the wavefunction and reading out the corresponding eigenvalues [45]. In practice, Eq. 1 is solved by expressing Ψ_{el} as an antisymmetric product of molecular orbitals

$$\psi_i = \sum_j C_{ij}\phi_j \,, \tag{2}$$

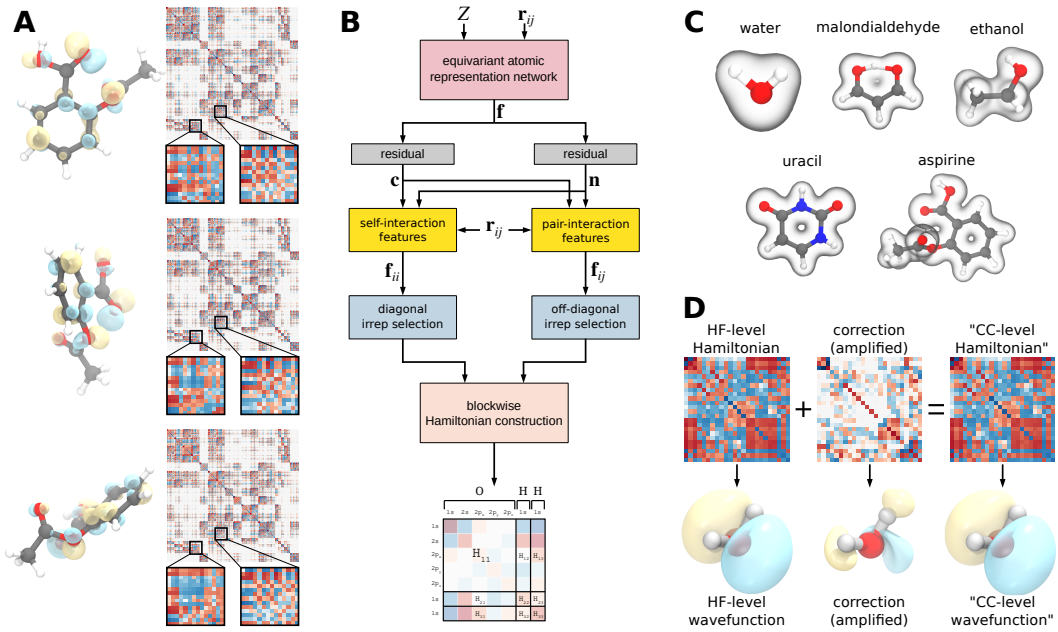

Figure 1: **A**: Illustration of an aspirine molecule and its highest occupied molecular orbital (HOMO) in three different orientations, showing how the wavefunction (left) and Hamiltonian matrix (right) change with respect to rotations. **B**: Overview of the PhiSNet architecture. The atomic representation network creates atom-wise equivariant features, which are used to produce self-interaction and pair-interaction features (Fig. S2), from which the Hamiltonian matrix is constructed block-by-block (Fig. S5). **C**: Visualisation of electronic densities (squared wavefunction) of various molecules predicted with our approach. **D**: Illustration of a transfer learning application, where a model pretrained on Hartree-Fock (HF) Hamiltonians is fine-tuned to match energies and forces derived from highly accurate coupled cluster (CC) calculations. The model achieves this by learning a correction to the Hamiltonian matrix, which mimics the effects of many-body electron correlation. The HOMO is shown to visualize subtle changes to the wavefunction.

which are written as linear combinations of atom-centered basis functions ϕ. This leads to the equation

$$\mathbf{HC} = \boldsymbol{\epsilon}\mathbf{SC}\,, \tag{3}$$

where the Hamiltonian is written as a matrix $\mathbf{H}$ with entries $H_{ij} = \int \phi_i^*(\mathbf{r})\hat{H}_{\mathrm{el}}\phi_j(\mathbf{r})d\mathbf{r}$ ($\mathbf{r}$ denotes the electronic coordinates). The overlap matrix $\mathbf{S}$ with entries $S_{ij} = \int \phi_i^*(\mathbf{r})\phi_j(\mathbf{r})d\mathbf{r}$ has to be introduced and a generalized eigenvalue problem must be solved, because the basis functions ϕ are typically not orthonormal. The eigenvectors $\mathbf{C}$ specify the wavefunction Ψ_{el} via the coefficients C_{ij} of the molecular orbitals (Eq. 2) and the eigenvalues $\boldsymbol{\epsilon}$ are the corresponding orbital energies. Since eigenvectors are only defined up to sign changes, predicting $\mathbf{H}$ (instead of $\mathbf{C}$) is preferable for machine learning applications. This is a challenging task, because the basis functions are typically products of a radial component and spherical harmonics, which introduces non-trivial dependencies of the matrix elements with respect to the orientation of the chemical system (see Fig. 1A). Spherical harmonics Y_l^m of degree $l = 0, \ldots, \infty$ and order $m = -l, \ldots, l$ form a complete orthonormal basis for functions on the surface of a sphere and can be used to derive irreducible representations (irreps) of the 3D rotation group $\mathrm{SO}(3)$. In other words, they are rotationally equivariant, which means that when $\mathbf{r}$ is rotated, the values of $Y_l^m(\mathbf{r})$ change accordingly. Since Ψ_{el} is expressed with spherical harmonic basis functions, the entries of $\mathbf{H}$ transform predictably under rotations (a more detailed overview of quantum chemistry fundamentals, groups, equivariance, and spherical harmonics is given in Section A of the supplement).

4 Deep learning architecture for molecular wavefunctions

Deep message-passing neural networks (MPNNs) [46] for quantum chemistry applications, such as DTNN [18] or SchNet [19], model physical properties of chemical systems as a sum over atomic contributions predicted from features $\mathbf{x}_i \in \mathbb{R}^F$ for each atom i. Starting from initial element-specific

embeddings, the features are constructed by iteratively exchanging "messages" between neighboring atoms i and j, which depend on the current $\mathbf{x}_i$ and $\mathbf{x}_j$ and their distance r_{ij}. Since geometric information enters only in the form of pairwise distances, the final atomic features are rotationally invariant by construction. This is desirable when they are used to predict a quantity that itself is rotationally invariant, for example the potential energy. To predict observables that change under rotation, e.g. electric moments or the electronic Hamiltonian, a natural extension is to instead construct rotationally equivariant features. To see how this can be achieved, it is useful to think of the F entries of atomic feature vectors $\mathbf{x}_i \in \mathbb{R}^F$ as different "channels", where each channel carries scalar information about the chemical environment of atom i. To construct rotationally equivariant features, each scalar channel can be replaced by values derived from the spherical harmonics up to a maximum degree L, i.e. there are now $F \times (L+1)^2$ entries (each spherical harmonic degree $l \in \{0, \dots, L\}$ contributes $2l+1$ values for all possible orders $m \in \{-l, \dots, l\}$). In the following, we describe novel general-purpose operations for building SE(3)-equivariant MPNNs, which can also be used outside a chemical context to build feature representations for other point cloud data. We then apply these building blocks to extend the design of the PhysNet architecture [23] to produce equivariant atomic representations. Our proposed model, which we call PhiSNet, constructs the Hamiltonian matrix in a block-wise manner from the atomic features.

Notation Whenever equivariant features are discussed, bold symbols (e.g. $\mathbf{x}$) refer to the collection of all $F \times (L+1)^2$ entries for all feature channels F and spherical harmonics degrees $l \in \{0, \dots, L\}$, whereas a superscript l in parentheses (e.g. $\mathbf{x}^{(l)}$) is used to refer only to the $F \times (2l+1)$ entries of degree l. Similarly, $\mathbf{Y}(\mathbf{r})$ refers to the collection of all $1 \times (L+1)^2$ spherical harmonics with distinct combinations of l and m, whereas $\mathbf{Y}^{(l)}(\mathbf{r})$ refers to the $1 \times (2l+1)$ values for degree l. The notation $\mathbf{c} \circ \mathbf{x}$ denotes a Hadamard product between matrices, i.e. $\mathbf{c} \in \mathbb{R}^{F \times (L+1)^2}$ and $\mathbf{x} \in \mathbb{R}^{F \times (L+1)^2}$ are multiplied entry-wise. When there is no one-to-one correspondence between the entries in $\mathbf{c}$ and $\mathbf{x}$, e.g. when $\mathbf{c} \in \mathbb{R}^F$ and $\mathbf{x} \in \mathbb{R}^{F \times (L+1)^2}$, $\mathbf{c} \circ \mathbf{x}$ implies that $\mathbf{c}$ is "broadcasted" across the missing dimensions, i.e. each of the $(L+1)^2$ "slices" of $\mathbf{x}$ is multiplied entry-wise with $\mathbf{c}$ and the result $\mathbf{c} \circ \mathbf{x}$ has dimensions $F \times (L+1)^2$. Double-struck digits denote an irreducible representation (irrep) with the corresponding number of dimensions $(2l+1)$, for example, $\mathbb{1}$ refers to a (one-dimensional) irrep of degree $l = 0$ and $\mathbb{3}$ to a (three-dimensional) irrep of degree $l = 1$. When a collection of equivariant features $\mathbf{x}^l \in \mathbb{R}^{F_{\text{in}} \times (2l+1)}$ is multiplied by a matrix $\mathbf{M} \in \mathbb{R}^{F_{\text{out}} \times F_{\text{in}}}$, the result is $\mathbf{M}\mathbf{x}^{(l)} \in \mathbb{R}^{F_{\text{out}} \times (2l+1)}$, i.e. the ordinary rules for matrix multiplication apply.

4.1 SE(3)-equivariant neural network building blocks

Activation functions may only be applied to scalar ($l = 0$) features, or else the output loses its equivariant properties:

$$\boldsymbol{\sigma}(\mathbf{x})^{(l)} = \begin{cases} \sigma(\mathbf{x}^{(l)}) & l = 0 \\ \mathbf{x}^{(l)} & l > 0 \end{cases} . \tag{4}$$

Here, σ can be any activation function and the notation $\sigma(\mathbf{x}^{(l)})$ means that σ is applied to $\mathbf{x}^{(l)}$ entry-wise. In this work, a generalized SiLU [47, 48] activation function (also known as Swish [49]) given by

$$\sigma(x) = \frac{\alpha x}{1 + e^{-\beta x}} \tag{5}$$

is used, where α and β are both learned and separate parameters are kept for all feature channels and instances of σ [27] (see Section B.1 in the supplement for additional details).

Linear layers are applied to each degree l according to

$$\text{linear}_{F_{\text{in}} \to F_{\text{out}}}(\mathbf{x})^{(l)} = \begin{cases} \mathbf{W}_l \mathbf{x}^{(l)} + \mathbf{b} & l = 0 \\ \mathbf{W}_l \mathbf{x}^{(l)} & l > 0 \end{cases} , \tag{6}$$

where $\mathbf{W} \in \mathbb{R}^{F_{\text{out}} \times F_{\text{in}}}$ and $\mathbf{b} \in \mathbb{R}^{F_{\text{out}}}$ are weights and biases, respectively. The subscript l is used to distinguish the weights for different degrees l, i.e. separate linear transformations are applied to the features of each degree l. The bias term must be omitted for $l > 0$ so that output features stay rotationally equivariant.

Tensor product contractions are used to couple two equivariant feature representations $\mathbf{x}^{(l_1)}$ and $\mathbf{y}^{(l_2)}$ to form new features $\mathbf{z}^{(l_3)}$. The (reducible) tensor product $\mathbf{x}^{(l_1)} \otimes \mathbf{y}^{(l_2)}$ of two irreps has $(2l_1+1)(2l_2+1)$ dimensions and can be expanded into a direct sum of irreducible representations, e.g. $\mathbb{3} \otimes \mathbb{5} = \mathbb{3} \oplus \mathbb{5} \oplus \mathbb{7}$. In general, the value for order m_3 of the irrep of degree $l_3 \in \{|l_1 - l_2|, \ldots, l_1 + l_2\}$ in the direct sum representation of the tensor product $\mathbf{x}^{(l_1)} \otimes \mathbf{y}^{(l_2)}$ is given by

$$\left(\mathbf{x}^{(l_1)} \otimes \mathbf{y}^{(l_2)}\right)^{l_3}_{m_3} = \sum_{m_1=-l_1}^{l_1} \sum_{m_2=-l_2}^{l_2} C^{l_3,l_2,l_1}_{m_3,m_2,m_1} x^{l_1}_{m_1} y^{l_2}_{m_2} \,, \tag{7}$$

where $C^{l_3,l_2,l_1}_{m_3,m_2,m_1}$ are Clebsch-Gordan coefficients (CGCs) [50]. The short-hand notation $\mathbf{x}^{(l_1)} \underset{l_3}{\otimes} \mathbf{y}^{(l_2)}$ is used to refer to the irrep of degree l_3 in the direct sum representation of $\mathbf{x}^{(l_1)} \otimes \mathbf{y}^{(l_2)}$. In other words, the operation $\mathbf{x}^{(l_1)} \underset{l_3}{\otimes} \mathbf{y}^{(l_2)}$ performs the tensor product and contracts the result to a single irrep of degree l_3.

Tensor product expansions are inverse to tensor product contractions. Instead of contracting two irreps into one, CGCs are used to expand a single irrep $\mathbf{x}^{(l_3)}$ into a $(2l_1+1) \times (2l_2+1)$ matrix that represents its contribution to the direct sum representation of the tensor product of two irreps of order l_1 and l_2, where $|l_2 - l_1| \leq l_3 \leq l_2 + l_1$:

$$\left(\overline{\otimes}\mathbf{x}^{(l_3)}\right)^{l_1,l_2}_{m_1,m_2} = \sum_{m_3=-l_3}^{l_3} \mathbf{C}^{l_1,l_2,l_3}_{m_l,m_2,m_3} x^{l_3}_{m_3} \,. \tag{8}$$

Mirroring the shorthand used for tensor product contractions, $\overset{l_1,l_2}{\overline{\otimes}} \mathbf{x}^{(l_3)}$ will be used to refer to the $(2l_1+1) \times (2l_2+1)$ matrix that is obtained from the tensor product expansion of $\mathbf{x}^{(l_3)}$.

Selfmix layers are used to recombine ("mix") the F features of a single input $\mathbf{x} \in \mathbb{R}^{F \times (L_{\text{in}}+1)^2}$ across different degrees and optionally allow changing the maximum degree from L_{in} to L_{out}. The output features of degree l_3 are given by

$$\text{selfmix}_{L_{\text{in}} \to L_{\text{out}}}(\mathbf{x})^{(l_3)} = \begin{cases} \mathbf{k}_{l_3} \circ \mathbf{x}^{(l_3)} + \sum_{l_1=0}^{L_{\text{in}}} \sum_{l_2=l_1+1}^{L_{\text{in}}} \mathbf{s}_{l_3,l_2,l_1} \circ \left(\mathbf{x}^{(l_1)} \underset{l_3}{\otimes} \mathbf{x}^{(l_2)}\right) & l_3 \leq L_{\text{in}} \\ \sum_{l_1=0}^{L_{\text{in}}} \sum_{l_2=l_1+1}^{L_{\text{in}}} \mathbf{s}_{l_3,l_2,l_1} \circ \left(\mathbf{x}^{(l_1)} \underset{l_3}{\otimes} \mathbf{x}^{(l_2)}\right) & l_3 > L_{\text{in}} \end{cases} . \tag{9}$$

Here, $\mathbf{k}, \mathbf{s} \in \mathbb{R}^F$ are learnable coefficients and the subscripts are used to distinguish independent parameters $\mathbf{k}, \mathbf{s}$ for different degrees: In total, a selfmix layer has $L_{\text{out}} + 1$ different $\mathbf{k}_{l_3}$ (one for each possible value of $l_3 \in \{0, \ldots, L_{\text{out}}\}$) and $(L_{\text{out}}+1)\frac{L_{\text{in}}(L_{\text{in}}+1)}{2}$ different $\mathbf{s}_{l_3,l_2,l_1}$ (one for each valid combination of l_3, l_2, l_1).

Spherical linear layers Spherical linear layers are a combination of linear (Eq. 6) and selfmix (Eq. 9) layers given by

$$\text{sphlinear}_{L_{\text{in}} \to L_{\text{out}}, F_{\text{in}} \to F_{\text{out}}}(\mathbf{x}) = \text{linear}_{F_{\text{in}} \to F_{\text{out}}}\left(\text{selfmix}_{\text{L}_{\text{in}} \to \text{L}_{\text{out}}}(\mathbf{x})\right) \,. \tag{10}$$

Chaining both operations allows arbitrary combinations across feature channels and degrees while still preserving rotational equivariance.

Residual blocks are modules consisting of two sequential spherical linear layers (see Eq. 10) and activation functions (see Eqs. 4 and 5) inspired by the pre-activation residual block described in [51]:

$$\text{residual}(\mathbf{x}) = \mathbf{x} + \text{sphlinear}_2(\boldsymbol{\sigma}_2(\text{sphlinear}_1(\boldsymbol{\sigma}_1(\mathbf{x})))) \,. \tag{11}$$

Here, $F_{\text{in}} = F_{\text{out}}$ and $L_{\text{in}} = L_{\text{out}}$ for both spherical linear layers.

Pairmix layers are used to combine a pair of features $\mathbf{x} \in \mathbb{R}^{F\times(L_x+1)^2}$ and $\mathbf{y} \in \mathbb{R}^{F\times(L_y+1)^2}$ with a scalar r (e.g. their Euclidean distance) to generate new features of degree L_{out}:

$$\text{pairmix}_{L_x,L_y\to L_{\text{out}}}(\mathbf{x},\mathbf{y},r)^{(l_3)} = \sum_{l_1=0}^{L_x}\sum_{l_2=0}^{L_y}\left(\mathbf{W}_{l_3,l_2,l_1}\mathbf{g}(r)\right)\circ\left(\mathbf{x}^{(l_1)}\underset{l_3}{\otimes}\mathbf{y}^{(l_2)}\right) . \tag{12}$$

Here, $\mathbf{g}(r) \in \mathbb{R}^K$ is the vector $[g_0(r)\ g_1(r)\ \dots\ g_{K-1}(r)]^\top$ and $g_k(r)$ are radial basis functions. In this work, exponential Bernstein polynomials [27] are used (see Eq. S9). The weight matrices $\mathbf{W} \in \mathbb{R}^{F\times K}$ allow to learn radial functions as linear combinations of the basis functions $g_k(r)$ and subscripts are used to distinguish independent weights for different combinations of l_1, l_2, l_3 (in total, there are $(L_x+1)(L_y+1)(L_{\text{out}}+1)$ possible combinations).

Interaction blocks use message-passing to model interactions between the features $\mathbf{c} \in \mathbb{R}^{F\times(L+1)^2}$ of a central point i with features $\mathbf{n} \in \mathbb{R}^{F\times(L+1)^2}$ of neighboring points j within a local environment:

$$\begin{aligned}
\mathbf{a}(\mathbf{x},\mathbf{r})^{(l)} &= \mathbf{x}^0 \circ (\mathbf{W}_l\mathbf{g}(\|\mathbf{r}\|)) \circ \text{sphlinear}_{L\to L,1\to F}(\mathbf{Y}(\mathbf{r}))^{(l)} ,\\
\mathbf{b}(\mathbf{x},\mathbf{r})^{(l)} &= \text{pairmix}_{L,L\to L}(\mathbf{x}, \text{sphlinear}_{L\to L,1\to F}\left(\mathbf{Y}(\mathbf{r})\right), \|\mathbf{r}\|)^{(l)} ,\\
\text{interaction}(\mathbf{c},\mathbf{n},\mathbf{r})_i^{(l)} &= \mathbf{c}_i^{(l)} + \sum_{j\neq i}\left(\mathbf{a}(\mathbf{n}_j,\mathbf{r}_{ij})^{(l)} + \mathbf{b}(\mathbf{n}_j,\mathbf{r}_{ij})^{(l)}\right) .
\end{aligned} \tag{13}$$

Here $\mathbf{r}_{ij}$ is the distance vector $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$ between the positions $\mathbf{r}_i, \mathbf{r}_j$ of i and j. The radial basis function expansion $\mathbf{g}$ is the same as in pairmix layers and $\mathbf{W} \in \mathbb{R}^{F\times K}$ are independent weight matrices for each order l. Since geometric information enters Eq. 13 via relative distance vectors $\mathbf{r}_{ij}$ expanded in a spherical harmonics basis, interactions blocks are equivariant with respect to the SE(3) group of roto-translations.

4.2 PhiSNet architecture

PhiSNet takes as inputs nuclear charges Z and positions $\mathbf{r}$ of N atoms, which are used to construct equivariant feature representations encoding information about the chemical environment of each atom. These features are then further transformed and used to predict the entries of the Hamiltonian matrix, see below. An overview over the complete architecture is shown in Fig. 1B and more detailed diagrams of individual building blocks are given in Fig. S2.

Atomic feature representations An embedding layer produces initial atomic feature representations $\mathbf{x}$ from the nuclear charges Z according to

$$\text{embedding}(Z)^{(l)} = \begin{cases}\mathbf{W}\mathbf{d}_Z + \mathbf{b}_Z & l = 0\\ \mathbf{0} & l > 0\end{cases} , \tag{14}$$

where $\mathbf{W} \in \mathbb{R}^{F\times 4}$ is a weight matrix and $\mathbf{b}_Z$ element-specific biases with learnable parameters. Here, $\mathbf{d}_Z \in \mathbb{R}^4$ are fixed vectors for each element that contain information about their nuclear charge and ground state electron configuration, similar to the embeddings described in [27] (see Section C.1 in the supplement for details). The features are then refined by five sequential modules, each consisting of identical building blocks with independent parameters:

$$\begin{aligned}
\mathbf{t} &= \text{residual}(\mathbf{x}) ,\\
\mathbf{i} &= \text{sphlinear}_{L\to L,F\to F}\left(\boldsymbol{\sigma}\left(\text{residual}(\mathbf{t})\right)\right) ,\\
\mathbf{j} &= \text{sphlinear}_{L\to L,F\to F}\left(\boldsymbol{\sigma}\left(\text{residual}(\mathbf{t})\right)\right) ,\\
\mathbf{v} &= \text{sphlinear}_{L\to L,F\to F}\left(\boldsymbol{\sigma}\left(\text{residual}(\text{interaction}(\mathbf{i},\mathbf{j},\mathbf{r}))\right)\right) ,\\
\tilde{\mathbf{x}} &= \text{residual}\left(\mathbf{t}+\mathbf{v}\right) ,\\
\tilde{\mathbf{y}} &= \text{residual}\left(\tilde{\mathbf{x}}\right) .
\end{aligned} \tag{15}$$

Each module produces two different outputs $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$. The first output $\tilde{\mathbf{x}}$ serves as input to the next module in the chain, whereas the second output $\tilde{\mathbf{y}}$ is summed with the outputs of other modules to form the final atomic feature representations $\mathbf{f} = \sum \tilde{\mathbf{y}}$.

Hamiltonian matrix prediction The Hamiltonian matrix is constructed block-by-block, with each block corresponding to the interaction between two atoms i and j. Diagonal and off-diagonal blocks are treated separately, i.e. different atomic pair features are constructed to predict them. Since these transformations also involve interactions with neighboring atoms (similar to interaction blocks), separate representations for central $\mathbf{c}$ and neighboring atoms $\mathbf{n}$ are created from the atomic features $\mathbf{f}$:

$$\begin{aligned} \mathbf{c} &= \mathrm{residual}(\mathbf{f})\,, \\ \mathbf{n} &= \mathrm{residual}(\mathbf{f})\,. \end{aligned} \tag{16}$$

Self-interaction features $\mathbf{f}_{ii}$ (for diagonal blocks) are constructed according to

$$\mathbf{f}_{ii} = \mathrm{residual}\left(\mathbf{c}_i^{(l)} + \sum_{j \neq i} \left(\mathbf{n}_j^{(l)} \circ \mathbf{W}_l \mathbf{g}(\|\mathbf{r}_{ij}\|)\right)\right)\,, \tag{17}$$

where $\mathbf{r}_{ij}$ is the distance vector between the positions of atoms i and j, and the radial basis function expansion $\mathbf{g}$ is the same as in $\mathrm{pairmix}$ layers. The matrices $\mathbf{W} \in \mathbb{R}^{F \times K}$ are separate trainable weight matrices for each order l.

Similarly, pair-interaction features $\mathbf{f}_{ij}$ (for off-diagonal blocks) are obtained by combining the central representations of atoms i and j and interacting them with the neighbors of atom i according to

$$\mathbf{f}_{ij} = \mathrm{residual}\left(\mathrm{pairmix}(\mathbf{c}_i, \mathbf{c}_j, \|\mathbf{r}_{ij}\|)^{(l)} + \sum_{k \notin \{i,j\}} \left(\mathbf{n}_k^{(l)} \circ \mathbf{W}_l \mathbf{g}(\|\mathbf{r}_{ik}\|)\right)\right)\,. \tag{18}$$

All blocks $\mathbf{H}_{ii}$ and $\mathbf{H}_{ij}$ of the Hamiltonian matrix, each representing the interaction between two atoms, are themselves composed of smaller blocks corresponding to the interaction between atomic orbitals. Since atomic orbitals are expressed in a spherical harmonics basis, their interactions transform non-trivially (put predictably) under rotations. The correct equivariant behavior of a matrix block $\mathbf{M}^{l_1,l_2} \in \mathbb{R}^{(2l_1+1)\times(2l_2+1)}$ corresponding to the interaction between orbitals of order l_1 and l_2 can be constructed as a sum over matrices obtained from tensor product expansions (Eq. 8) of irreps $\mathbf{a}$ of all valid orders $l_3 \in \{|l_2 - l_1|, \ldots, \leq l_2 + l_1\}$:

$$\mathbf{M}^{l_1,l_2} = \sum_{l_3=|l_2-l_1|}^{l_2+l_1} \overset{l_1,l_2}{\overline{\otimes}}\, \mathbf{a}^{(l_3)}\,. \tag{19}$$

Two sets of indices I^{self} and I^{pair} count and keep track of the irreps necessary to construct all required matrices $\mathbf{M}^{l_1,l_2}$. For diagonal blocks $\mathbf{H}_{ii}$, the irreps of a given degree l for the interaction of orbitals n and m of atoms with nuclear charge Z are collected from specific channels of the self-interaction features $\mathbf{f}_{ii}$ via a unique index $I^{\mathrm{self}}(Z, n, m, L)$. Similarly, for off-diagonal blocks $\mathbf{H}_{ij}$, a unique index $I^{\mathrm{pair}}(Z_i, Z_j, n_i, n_j, L)$ selects irreps corresponding to orbitals n_i and n_j of atom pairs with nuclear charges Z_i and Z_j from the pair-interaction features $\mathbf{f}_{ij}$. After all blocks have been constructed, the complete matrix $\tilde{\mathbf{H}}$ is obtained by placing individual blocks at the appropriate positions (based on which atoms and orbitals interact). Finally, the symmetric Hamiltonian matrix is constructed as $\mathbf{H} = \tilde{\mathbf{H}} + \tilde{\mathbf{H}}^T$. This symmetrization guarantees that both pair-features $\mathbf{f}_{ij}$ and $\mathbf{f}_{ji}$ contribute equally to the corresponding off-diagonal blocks and makes sure that sub-blocks of the Hamiltonian swap positions in the correct way when equivalent atoms are permuted. In cases where multiple Hamiltonian-like matrices need to be predicted, all parameters up to the final residual blocks in Eqs. 17 and 18 are shared. An exception are overlap matrices, for which simpler self- and pair-interaction features derived directly from the embeddings are sufficient (see Section C.3 in the supplement for details). The complete block-wise construction process of the Hamiltonian matrix from irreps is illustrated in Fig. S5 for a water molecule with minimal basis set.

5 Results and discussion

To assess the ability of PhiSNet to predict molecular wavefunctions and electronic densities (see Fig. 1C), we train it on Kohn-Sham and overlap matrices for various non-equilibrium configurations

Table 1: Prediction errors of PhiSNet for various molecules compared to SchNOrb [30]. In addition to Kohn-Sham $\mathbf{K}$ and overlap $\mathbf{S}$ matrices, we also report errors for energies ϵ and the cosine similarity between predicted and reference wavefunction ψ for all occupied orbitals. Best results in bold.

Data set		$\mathbf{K}$ $[10^{-6}\ \mathrm{E_h}]$	$\mathbf{S}$ $[10^{-6}]$	ϵ $[10^{-6}\ \mathrm{E_h}]$	ψ
Water	SchNOrb	165.4	79.1	279.3	**1.00**
	PhiSNet	**17.59**	**1.56**	**85.53**	**1.00**
Ethanol	SchNOrb	187.4	67.8	334.4	**1.00**
	PhiSNet	**12.15**	**0.626**	**62.75**	**1.00**
Malondialdehyde	SchNOrb	191.1	67.3	400.6	0.99
	PhiSNet	**12.32**	**0.567**	**73.50**	**1.00**
Uracil	SchNOrb	227.8	82.4	1760	0.90
	PhiSNet	**10.73**	**0.533**	**84.03**	**1.00**
Aspirin	SchNOrb	506.0	110	48689	0.57
	PhiSNet	**12.84**	**0.406**	**176.6**	**0.98**

of water, ethanol, malondialdehyde, uracil, and aspirin computed at the density functional theory (DFT) level with PBE/def2-SVP (the Kohn-Sham matrix takes the role of the Hamiltonian in DFT methods). Datasets for all molecules are taken from [30], with the exception of aspirin, for which geometries were sampled from the MD17 dataset [5] (more details on the datasets, training procedure, and hyperparameter settings can be found in Sections D and E of the supplement).

The results are summarized in Tab. 5 and compared to the current state-of-the-art given by SchNOrb [30]. PhiSNet achieves accuracy improvements up to two orders of magnitude, with the biggest differences arising in larger and more complex molecules like uracil and aspirin. Note that the training process for SchNOrb requires data augmentation via random rotations to approximate the equivariance relation between wavefunction and molecular orientation, while PhiSNet preserves exact equivariance. This not only allows for faster convergence, but also leads to much smaller model sizes, with our model requiring approximately one fifth of the parameters of SchNOrb while providing significantly more accurate results.

The improved prediction accuracy provided by PhiSNet makes it possible to accurately derive properties such as energies and forces directly from the wavefunction in an end-to-end manner, enabling a number of interesting and novel machine learning applications for the molecular sciences. As an example, we showcase a transfer-learning application, where a model trained on low accuracy Hartree-Fock (HF) electronic structure calculations is fine-tuned to learn a correction to the wavefunction, such that energies and forces match those obtained via high-level coupled cluster with singles, doubles, and perturbative triple excitations (CCSD(T)) calculations. The CCSD(T) method models the effects of electronic many-body interactions, which are neglected in HF theory, and is often considered to be the "gold standard" of quantum chemistry [8]. However, its accuracy comes at a significantly increased computational complexity, going from $\mathcal{O}(N^3)$ for HF to $\mathcal{O}(N^7)$ for CCSD(T) (here, N is the number of basis functions). Thus, a machine-learned correction to HF theory, which mimics the effects of electron correlation in a computationally efficient manner, is a possible way towards novel hybrid methods that rival the accuracy of high level electronic structure calculations and combine the generality and robustness of *ab initio* methods with the efficiency of machine learning.

After pretraining PhiSNet on HF/cc-pVDZ data for Fock matrices $\mathbf{F}$, core Hamiltonians $\mathbf{H}^{\mathrm{core}}$, and overlap matrices $\mathbf{S}$, we fine-tune it on forces computed at the CCSD(T)/cc-pVTZ level, but still keep loss terms for $\mathbf{H}^{\mathrm{core}}$ and $\mathbf{S}$ computed with HF/cc-pVDZ (see Section A.1 in the supplement for a brief overview of HF theory, where we explain the relevance of $\mathbf{F}$, $\mathbf{H}^{\mathrm{core}}$, and $\mathbf{S}$, and how energies and forces are derived from them). This way, the model learns to adapt only the Fock matrix $\mathbf{F}$, which embodies the electron-electron interactions in the HF formalism. Interestingly, very subtle changes to $\mathbf{F}$ seem to be sufficient to approximate the effects of electron correlation, resulting in an "effective CCSD(T) wavefunction" that, at first glance, appears to be almost identical to its original HF-level counterpart (see Fig. 1D). Nonetheless, the modified wavefunction reduces the mean absolute errors

(MAEs) between energies and forces predicted with PhiSNet to just 79 $\mu\mathrm{E_h}$ and 0.85 $\mathrm{mE_h a_0^{-1}}$, respectively, compared to the CCSD(T) reference. In contrast, the original HF-level wavefunction leads to MAEs of 4266 $\mu\mathrm{E_h}$ and 15 $\mathrm{mE_h a_0^{-1}}$ for energies and forces, respectively, i.e. prediction errors are reduced by over one order of magnitude with no additional computational overhead. More details on the transfer learning application can be found in Section E.2 of the supplement.

While the results are highly promising, the current approach also has limitations. Due to the fact that all pairwise combinations of atoms have to be considered for constructing the pair-interaction features $\mathbf{f}_{ij}$, the Hamiltonian matrix prediction scales quadratically with the number of atoms. Further, to derive the coefficient matrix $\mathbf{C}$ defining the wavefunction, a generalized eigenvalue problem has to be solved (see Eq. 3), which scales cubically with the matrix size (the number of basis functions). For these reasons, PhiSNet does not scale well to systems with a very large number of atoms. However, possible extensions of our method could exploit the fact that orbital overlaps between distant atoms are very small, so the entries of corresponding matrix blocks are approximately zero [52]. Then, only pair-interaction features for non-zero blocks need to be computed and the solution of Eq. 3 can exploit sparsity.

6 Conclusion

For learning problems with known invariances or equivariances, as is common in physics applications, it is useful to include such properties directly into the model architecture. Generally, this increases model performance and decreases the required amount of training data. While invariance and equivariance properties could also be (approximately) learned, this typically requires much more reference data and/or data augmentation. In contrast, by "hard coding" domain knowledge, the learning is constrained to a meaningful submanifold, e.g. reflecting rotational equivariance [35], energy conservation, physical laws or symmetries (e.g. [4, 5, 23, 27]), group equivariance (e.g. [34]), graph properties (e.g. [37, 43]) or alike. Thus, known properties do not need to be learned explicitly, because the submanifold where learning takes place already embodies them.

Our present contribution follows this design principle, specifically, we describe a series of general SE(3)-equivariant operations and building blocks for deep learning architectures operating on geometric point cloud data, which we used here to construct PhiSNet, a novel neural network architecture capable of accurately predicting wavefunctions and electronic densities. Unlike previous models, which need to approximately learn how the wavefunction transforms under molecular rotations and rely on data augmentation, the SE(3)-equivariant building blocks of our network allow to *exactly* capture the correct transformation without needing to learn it explicitly. By applying PhiSNet on a range of small to medium-sized molecules, we demonstrated that our model achieves accuracy improvements of up to *two orders of magnitude* compared to the previous state-of-the-art, while at the same time requiring significantly less parameters.

For the first time, sufficient accuracy is reached to derive quantum mechanical observations directly from the predicted wavefunction in an end-to-end manner, which also allows to adapt the predicted wavefunction such that it leads to desired physical properties. To showcase such an application, we fine-tuned a model trained on low accuracy wavefunctions to predict properties computed at a much higher level of theory, thereby learning a correction that implicitly captures the effects of many-body electron correlation. This paves the way towards the development of novel semi-empirical methods that are capable of providing highly accurate quantum chemical calculations at a drastically reduced computational cost.

Although we focus on quantum chemistry applications in this contribution, we would like to reiterate that the presented SE(3)-equivariant operations are general and can be used to construct other deep learning architectures for geometric point cloud data beyond physics, e.g. [34, 35, 37, 42, 43].

References

[1] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters*, 98(14):146401, 2007.

[2] Albert P. Bartók, Mike C. Payne, Risi Kondor, and Gábor Csányi. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Physical Review Letters*, 104(13):136403, 2010.

[3] Justin S. Smith, Olexandr Isayev, and Adrian E. Roitberg. ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chemical science*, 8(4):3192–3203, 2017.

[4] Stefan Chmiela, Alexandre Tkatchenko, Huziel E. Sauceda, Igor Poltavsky, Kristof T. Schütt, and Klaus-Robert Müller. Machine learning of accurate energy-conserving molecular force fields. *Science Advances*, 3(5):e1603015, 2017.

[5] Stefan Chmiela, Huziel E. Sauceda, Klaus-Robert Müller, and Alexandre Tkatchenko. Towards exact molecular dynamics simulations with machine-learned force fields. *Nature Communications*, 9(1):3887, 2018.

[6] Anders S. Christensen, Felix A. Faber, and O. Anatole von Lilienfeld. Operators in quantum machine learning: Response properties in chemical space. *J. Phys. Chem.*, 150(6):064105, 2019.

[7] Simon Batzner, Tess E. Smidt, Lixin Sun, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, and Boris Kozinsky. Se(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials, 2021.

[8] Oliver T Unke, Stefan Chmiela, Huziel E Sauceda, Michael Gastegger, Igor Poltavsky, Kristof T Schütt, Alexandre Tkatchenko, and Klaus-Robert Müller. Machine learning force fields. *Chemical Reviews*, 2021.

[9] John C Snyder, Matthias Rupp, Katja Hansen, Klaus-Robert Müller, and Kieron Burke. Finding density functionals with machine learning. *Physical review letters*, 108(25):253002, 2012.

[10] Felix Brockherde, Leslie Vogt, Li Li, Mark E. Tuckerman, Kieron Burke, and Klaus-Robert Müller. Bypassing the kohn-sham equations with machine learning. *Nature Communications*, 8: 872, 2017.

[11] Kevin Ryczko, David Strubbe, and Isaac Tamblyn. Deep learning and density functional theory. *arXiv preprint arXiv:1811.08928*, 2018.

[12] Andrea Grisafi, David M. Wilkins, Benjamin A.R. Meyer, Alberto Fabrizio, Clemence Corminboeuf, and Michele Ceriotti. A transferable machine-learning model of the electron density. *arXiv preprint arXiv:1809.05349*, 2018.

[13] Ryo Nagai, Ryosuke Akashi, and Osamu Sugino. Completing density functional theory by machine learning hidden messages from molecules. *npj Computational Materials*, 6(1):1–8, 2020.

[14] Mihail Bogojeski, Leslie Vogt-Maranto, Mark E Tuckerman, Klaus-Robert Müller, and Kieron Burke. Quantum chemical accuracy from density functional approximations via machine learning. *Nature communications*, 11:5223, 2020.

[15] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O. Anatole von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.*, 108(5):058301, 2012.

[16] Grégoire Montavon, Matthias Rupp, Vivekanand Gobre, Alvaro Vazquez-Mayagoitia, Katja Hansen, Alexandre Tkatchenko, Klaus-Robert Müller, and O. Anatole von Lilienfeld. Machine learning of molecular electronic properties in chemical compound space. *New J. Phys.*, 15(9): 095003, 2013.

[17] Sandip De, Albert P Bartók, Gábor Csányi, and Michele Ceriotti. Comparing molecules and solids across structural and alchemical space. *Physical Chemistry Chemical Physics*, 18(20): 13754–13769, 2016.

[18] Kristof T Schütt, Farhad Arbabzadah, Stefan Chmiela, Klaus-Robert Müller, and Alexandre Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nature Communications*, 8:13890, 2017.

[19] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in Neural Information Processing Systems*, pages 991–1001, 2017.

[20] 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.

[21] Wiktor Pronobis, Alexandre Tkatchenko, and Klaus-Robert Müller. Many-body descriptors for predicting molecular properties with machine learning: Analysis of pairwise and three-body interactions in molecules. *J. Chem. Theory Comput.*, 14(6):2991–3003, 2018.

[22] Michael Eickenberg, Georgios Exarchakis, Matthew Hirn, Stéphane Mallat, and Louis Thiry. Solid harmonic wavelet scattering for predictions of molecule properties. *J. Chem. Phys.*, 148 (24):241732, 2018.

[23] Oliver T Unke and Markus Meuwly. Physnet: A neural network for predicting energies, forces, dipole moments, and partial charges. *J. Chem. Theory Comput.*, 15(6):3678–3693, 2019.

[24] O Anatole von Lilienfeld, Klaus-Robert Müller, and Alexandre Tkatchenko. Exploring chemical compound space with quantum-based machine learning. *Nat. Rev. Chem.*, 4:347–358, 2020.

[25] Zhantao Chen, Nina Andrejevic, Tess Smidt, Zhiwei Ding, Qian Xu, Yen-Ting Chi, Quynh T. Nguyen, Ahmet Alatas, Jing Kong, and Mingda Li. Direct prediction of phonon density of states with euclidean neural networks. *Advanced Science*, page 2004214, March 2021. doi: 10.1002/advs.202004214. URL `https://doi.org/10.1002/advs.202004214`.

[26] Kristof T Schütt, Oliver T Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. *arXiv preprint arXiv:2102.03150*, 2021.

[27] Oliver T Unke, Stefan Chmiela, Michael Gastegger, Kristof T Schütt, Huziel E Sauceda, and Klaus-Robert Müller. Spookynet: Learning force fields with electronic degrees of freedom and nonlocal effects. *arXiv preprint arXiv:2105.00304*, 2021.

[28] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

[29] Ganesh Hegde and R Chris Bowen. Machine-learned approximations to density functional theory hamiltonians. *Scientific reports*, 7:42669, 2017.

[30] KT 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. *Nat. Commun.*, 10(1):1–10, 2019.

[31] He Li, Zun Wang, Nianlong Zou, Meng Ye, Wenhui Duan, and Yong Xu. Deep neural network representation of density functional theory hamiltonian. *arXiv preprint arXiv:2104.03786*, 2021.

[32] Grégoire Montavon, Katja Hansen, Siamac Fazli, Matthias Rupp, Franziska Biegler, Andreas Ziehe, Alexandre Tkatchenko, Anatole Lilienfeld, and Klaus-Robert Müller. Learning invariant representations of molecules for atomization energy prediction. *Advances in neural information processing systems*, 25:440–448, 2012.

[33] M Gastegger, A McSloy, M Luya, KT Schütt, and RJ Maurer. A deep neural network for molecular wave functions in quasi-atomic minimal basis representation. *J. Chem. Phys.*, 153(4): 044123, 2020.

[34] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016.

[35] Diego Marcos, Michele Volpi, Nikos Komodakis, and Devis Tuia. Rotation equivariant vector field networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5048–5057, 2017.

[36] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018.

[37] Risi Kondor, Hy Truong Son, Horace Pan, Brandon Anderson, and Shubhendu Trivedi. Covariant compositional networks for learning graphs. *arXiv preprint arXiv:1801.02144*, 2018.

[38] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. *arXiv preprint arXiv:1807.02547*, 2018.

[39] 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.

[40] Andrea Grisafi, David M. Wilkins, Gábor Csányi, and Michele Ceriotti. Symmetry-adapted machine learning for tensorial properties of atomistic systems. *Phys. Rev. Lett.*, 120:036002, Jan 2018. doi: 10.1103/PhysRevLett.120.036002.

[41] Truong Son Hy, Shubhendu Trivedi, Horace Pan, Brandon M. Anderson, and Risi Kondor. Predicting molecular properties with covariant compositional networks. *J. Chem. Phys.*, 148 (24):241745, 2018.

[42] Taco S Cohen, Mario Geiger, and Maurice Weiler. A general theory of equivariant cnns on homogeneous spaces. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[43] Nicolas Keriven and Gabriel Peyré. Universal invariant and equivariant graph neural networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[44] Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral cnn. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2019.

[45] Martin Schechter. *Operator methods in quantum mechanics*. Elsevier, 2012.

[46] 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.

[47] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[48] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Netw.*, 107:3–11, 2018.

[49] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

[50] D A Varshalovich, A N Moskalev, and V K Khersonskii. *Quantum theory of angular momentum*. World Scientific, 1988.

[51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[52] Stefan Goedecker and GE Scuserza. Linear scaling electronic structure methods in chemistry and physics. *Computing in Science & Engineering*, 5(4):14–21, 2003.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] The limitations of our work are discussed in Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [No] We could not identify any potential negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code and instructions are included in the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] All training details are described in the supplemental material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A] We test the robustness of our method by applying it to five different molecular datasets, but we did not repeat experiments on a single dataset multiple times.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We give details on the Hardware we ran our experiments on in the supplemental material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite Schütt et al. [30] and Chmiela et al. [5].
   (b) Did you mention the license of the assets? [N/A] The datasets we use are not published under a specific license, but are freely available from `http://www.quantum-machine.org/datasets/`.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] All datasets we generated ourselves are included in the supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] All data we use is freely available online for public use.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] Only quantum chemistry data is used in this work.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]