So3krates: Equivariant attention for interactions on arbitrary length-scales in molecular systems

J. Thorben Frank[1,2]* **Oliver T. Unke**[1,2,3]

Klaus-Robert Müller[1,2,3,4,5]†

[1] Machine Learning Group, TU Berlin, 10587 Berlin, Germany
[2] BIFOLD, Berlin Institute for the Foundations of Learning and Data, Germany
[3] Google Research, Brain team, Berlin
[4] Department of Artificial Intelligence, Korea University, Seoul 136-713, Korea
[5] Max Planck Institut für Informatik, 66123 Saarbrücken, Germany

Abstract

The application of machine learning methods in quantum chemistry has enabled the study of numerous chemical phenomena, which are computationally intractable with traditional *ab-initio* methods. However, some quantum mechanical properties of molecules and materials depend on non-local electronic effects, which are often neglected due to the difficulty of modeling them efficiently. This work proposes a modified attention mechanism adapted to the underlying physics, which allows to recover the relevant non-local effects. Namely, we introduce *spherical harmonic coordinates* (SPHCs) to reflect higher-order geometric information for each atom in a molecule, enabling a non-local formulation of attention in the SPHC space. Our proposed model SO3KRATES[3] – a self-attention based message passing neural network – uncouples geometric information from atomic features, making them independently amenable to attention mechanisms. Thereby we construct *spherical filters*, which extend the concept of continuous filters in Euclidean space to SPHC space and serve as foundation for a *spherical self-attention* mechanism. We show that in contrast to other published methods, SO3KRATES is able to describe non-local quantum mechanical effects over arbitrary length scales. Further, we find evidence that the inclusion of higher-order geometric correlations increases data efficiency and improves generalization. SO3KRATES matches or exceeds state-of-the-art performance on popular benchmarks, notably, requiring a significantly lower number of parameters (0.25–0.4x) while at the same time giving a substantial speedup (6–14x for training and 2–11x for inference) compared to other models.

1 Introduction

Atomistic simulations use long time-scale molecular dynamics (MD) trajectories to predict macroscopic properties that arise from interactions on the microscopic scale [1–3]. Their predictive reliability is determined by the accuracy of the underlying *force field* (FF), which needs to be queried at every time step. This quickly becomes a computational bottleneck if the forces are determined from first principles, which may be required for accurate results. To that end, machine learning

*thorbenjan.frank@googlemail.com
†klaus-robert.mueller@tu-berlin.de
[3] `https://github.com/thorben-frank/mlff`

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

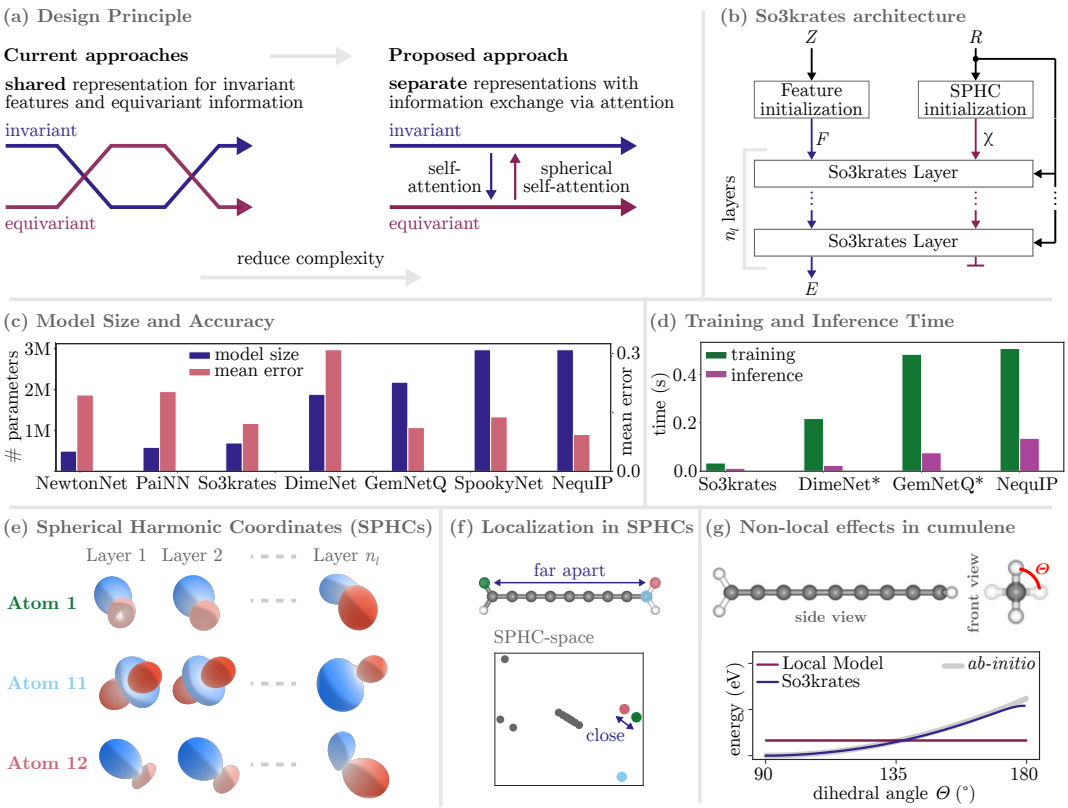

Figure 1: **(a)** Design principle of our proposed message-passing scheme, were invariant features $\mathbf{f}$ and equivariant information - represented by *spherical harmonic coordinates* (SPHCs) $\boldsymbol{\chi}$ - are separate and exchange information via (spherical) self-attention. **(b)** Overview of the So3krates architecture. **(c)** Comparison of prediction accuracy and model size of different architectures. **(d)** Comparison of training and inference times of different architectures. **(e)** Illustration of learned SPHCs at different layers of the So3krates architecture. **(f)** Low-dimensional projection of atomic SPHCs, showing that atoms far apart in Euclidean space can be mapped close together in SPHC space. **(g)** In contrast to local models, So3krates is able to learn the long-range correlations between hydrogen rotors in cumulene and reproduces the *ab-initio* ground truth faithfully.

FFs (MLFFs) offer a computationally more efficient, yet accurate empirical alternative to expensive *ab-initio* methods [2, 4–24].

In recent years, *Geometric Deep Learning* has become a popular design paradigm, which exploits relevant symmetry groups of the underlying learning problem by incorporating a *geometric prior* [12, 25, 26]. This effectively restricts the learnable functions of the model to a subspace with a meaningful inductive bias. Prominent examples for such models are e.g. convolutional neural networks (CNNs) [27], which are equivariant w. r. t. the group of translations, or graph neural networks (GNNs) [28], which are invariant w. r. t. node permutation.

For molecular property prediction, it has been shown that equivariance w. r. t. the 3D rotation group $\mathrm{SO}(3)$ greatly improves data efficiency and accuracy of the learned FFs [29–32]. To achieve equivariance, architectures either rely on feature expansions in terms of spherical harmonics (SH) [33] or explicitly include (dihedral) angles [29, 34]. While the latter scales quadratically (cubically) in the number of neighboring atoms and has been shown to be geometrically incomplete [35], the calculation of spherical harmonics scales only linear in the number of neighboring atoms, which makes them a fast and accurate alternative [30, 32, 36].

[*] Reference times were taken from [34]. As our own timings were measured on a different GPU, we decreased the reported times according to speedup-factors reported in [37]. For full details, see appendix A.6.

However, current higher-order geometric representations based on SHs usually result in expensive transformations, since an individual feature channel per SH degree (and order) is required. As a result, going to higher degrees is computationally expensive and comes at the price of increasing complexity, resulting in state-of-the-art (SOTA) models with millions of parameters [30, 32, 34]. However, in order to be applicable to large molecular structures, models are required to be both efficient and accurate on *all* length scales.

Non-local electronic effects have been outlined as one of the major challenges for a new generation of MLFFs [21]. They result in non-local, higher-order geometric relations between atoms. Most current architectures implicitly assume locality of interactions (expressed through a local neighborhood), which prohibits an efficient description of all relevant atomic interactions at larger scales. Simply increasing the cutoff radius used to determine local neighborhoods is not an adequate solution, since it only shifts the problem to larger length scales [30].

In this work, we propose *spherical harmonic coordinates* (SPHCs), which encode higher-order geometric information for each node in a molecular graph (Fig. 1e). This is in stark contrast to current approaches, which consider molecules as three-dimensional point clouds with learned features and fixed atomic coordinates: we propose to make the SPHCs themselves a *learned* quantity. Through localization in the space of SPHCs (Fig. 1f), models are able to efficiently describe electronic effects that are non-local in three-dimensional Euclidean space (Fig. 1g).

We then present SO3KRATES (Fig. 1b), a self-attention based message-passing neural network (MPNN), which decouples atomic features and SPHCs and updates them individually (Fig. 1a). This resembles ideas from equivariant graph neural networks [25], but allows to go to arbitrarily high geometric orders. The separation of higher-order geometric and feature information allows to overcome the parametric and computational complexity usually encountered in models with higher-order geometric representations, since we only require a single feature channel (instead of one per SH degree and order). Thus, SO3KRATES resembles some early architectures like SCHNET [10] or PHYSNET [19] in parametric simplicity. We further show that SO3KRATES outperforms the popular sGDML [38] kernel model by a large margin in the low-data regime, a domain which has so far been considered to be dominated by kernel machines [21]. Numerical evidence suggests that the data efficiency of SO3KRATES is directly related to the maximal degree of geometric information encoded in the SPHCs. We then apply SO3KRATES to the well-established MD17 benchmark and show that our model achieves SOTA results, despite is light-weight structure and having only 0.25–0.4x the number of parameters of competitive architectures (Fig. 1c), while achieving speedups of 6–14x and 2–11x for training and inference, respectively (Fig. 1d).

Although we focus on quantum chemistry applications in this work, the developed methods are also applicable to other fields where long-ranged correlations in three-dimensional data are relevant. For example, models based on SPHCs may also be applicable to tasks like 3D shape classification or computer vision.

2 Preliminaries and Related Work

In the following, we review the most important concepts our method is based upon and relate it to prior work.

Message Passing Neural Networks MPNNs [14] carry over many of the benefits of convolutions to unstructured domains and have thus become one of the most promising approaches for the description of molecular properties. Their general working principle relies on the repeated iteration of message passing (MP) steps, which can be phrased as follows [14, 25]

$$\mathbf{m}_{ij} = m(\mathbf{f}_i, \mathbf{f}_j, \boldsymbol{r}_{ij}) \tag{1}$$

$$\mathbf{m}_i = \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij} \tag{2}$$

$$\mathbf{f}_i' = u(\mathbf{f}_i, \mathbf{m}_i)\,. \tag{3}$$

Here, $\mathbf{m}_{ij}$ is the message between atoms i and j computed with the message function $m(\cdot)$, $\mathbf{m}_i$ is the aggregation of all messages in the neighborhood $\mathcal{N}(i)$ of atom i, and $u(\cdot)$ is an update function returning updated features $\mathbf{f}_i'$ based on the current features $\mathbf{f}_i$ and message $\mathbf{m}_i$. The neighborhood

$\mathcal{N}(i)$ consists of all atoms which lie within a given cutoff radius around the atomic position $\boldsymbol{r}_i$, which ensures linear scaling in the number of atoms n. While earlier variants parametrized messages only in terms of inter-atomic distances [13, 19], more recent approaches also take higher-order geometric information into account [25, 29, 31, 32, 39, 40].

Molecules as Point Clouds A molecule can be considered as a point cloud of n atoms $\mathcal{P}_{3D}(\mathcal{R}, \mathcal{F})$, where $\mathcal{R} = (\boldsymbol{r}_1, ...\boldsymbol{r}_n)$ denotes the set of atomic positions $\boldsymbol{r}_i \in \mathbb{R}^3$ and $\mathcal{F} = (\mathbf{f}_1, \ldots, \mathbf{f}_n)$ is the set of rotationally invariant atomic descriptors, or features, $\mathbf{f}_i \in \mathbb{R}^F$. We write the distance vector pointing from the position of atom i to the position of atom j as $\boldsymbol{r}_{ij} = \boldsymbol{r}_j - \boldsymbol{r}_i$, the distance as $r_{ij} = \|\boldsymbol{r}_{ij}\|_2$ and the normalized distance vector as $\hat{\boldsymbol{r}}_{ij} = \boldsymbol{r}_{ij}/r_{ij}$. Given the point cloud, a density over Euclidean space assigning a vector value to each point $\boldsymbol{r}$ can be constructed as

$$\boldsymbol{\rho}(\boldsymbol{r}) = \sum_{i=1}^{n} \delta(\|\boldsymbol{r}_i - \boldsymbol{r}\|_2) \cdot \mathbf{f}_i \,, \tag{4}$$

where δ is the Dirac delta function. It can be shown that applying a convolutional filter on $\boldsymbol{\rho}(\boldsymbol{r})$ resembles the update steps used in MPNNs [41].

Equivariance Given a set of transformations that act on a vector space $\mathbb{A}$ as $S_g : \mathbb{A} \mapsto \mathbb{A}$ to which we associate an abstract group G, a function $f : \mathbb{A} \mapsto \mathbb{B}$ is said to be equivariant w. r. t. G if

$$f(S_g x) = T_g f(x) \,, \tag{5}$$

where $T_g : \mathbb{B} \mapsto \mathbb{B}$ is an equivalent transformation on the output space [25]. Thus, in order to say that f is equivariant, it must hold that under transformation of the input, the output transforms "in the same way". While equivariance has been a popular concept in signal processing for decades (cf. e.g. [42] or wavelet neural networks [43]), recent years have seen efforts to design group equivariant NNs and kernel methods, since respecting relevant symmetries builds an important inductive bias [12, 44, 45]. Examples are CNNs [27] which are equivariant w. r. t. translation, GNNs [14, 28] which are invariant ($T_g = \mathbb{I}$) w. r. t. permutation, or architectures which are equivariant w. r. t. the $\mathrm{SO}(3)$ group [25, 31, 33, 36, 46]. In this work, we consider the $\mathrm{SO}(3)$ group of rotations, such that $\mathbb{A}$ is the Euclidean space $\mathbb{R}^3$, where the corresponding group actions are given by rotation matrices $R_\theta \in \mathbb{R}^{3\times 3}$.

Spherical Harmonics The spherical harmonics are special functions defined on the surface of the sphere $S^2 = \{\hat{\boldsymbol{r}} \in \mathbb{R}^3 : \|\hat{\boldsymbol{r}}\|_2 = 1\}$ and form an orthonormal basis for the irreducible representations (irreps) of $\mathrm{SO}(3)$. In the context of *tensor field networks* [33], they have been introduced as elementary building blocks for $\mathrm{SO}(3)$-equivariant neural networks. The spherical harmonics are commonly denoted as $Y_l^m(\hat{\boldsymbol{r}}) : S^2 \mapsto \mathbb{R}$, where the *degree* l determines all possible values of the *order* $m \in \{-l, \ldots, +l\}$. They transform under rotation as

$$Y_l^m(R_\theta\, \hat{\boldsymbol{r}}) = \sum_{m'} D^l_{mm'}(R_\theta)\, Y_l^{m'}(\hat{\boldsymbol{r}}) \,, \tag{6}$$

where $D^l_{mm'}(R_\theta)$ are the entries of the *Wigner-D* matrix $\mathbf{D}^l(R_\theta) \in \mathbb{R}^{(2l+1)\times(2l+1)}$ [47]. Based on the spherical harmonics, we define a vector-valued function $\boldsymbol{Y}^{(l)} : S^2 \mapsto \mathbb{R}^{2l+1}$ for each degree l, with entries Y_l^m for all valid orders m of a given degree l. Since $\boldsymbol{Y}^{(l)}(R_\theta\, \hat{\boldsymbol{r}}) = \mathbf{D}^l(R_\theta)\boldsymbol{Y}^{(l)}(\hat{\boldsymbol{r}})$ (cf. eq. (6)), $\boldsymbol{Y}^{(l)}$ is equivariant w. r. t. $\mathrm{SO}(3)$.

Tensor Product Contractions The irreps $\boldsymbol{Y}^{(l_1)}$ and $\boldsymbol{Y}^{(l_2)}$ can be coupled by computing their tensor product $\boldsymbol{Y}^{(l_1)} \otimes \boldsymbol{Y}^{(l_2)}$, which can equivalently be expressed as a direct sum [33, 48]

$$\boldsymbol{Y}^{(l_1)} \otimes \boldsymbol{Y}^{(l_2)} = \bigoplus_{l_3=|l_1-l_2|}^{l_1+l_2} \overbrace{\tilde{\boldsymbol{Y}}^{(l_3)}}^{:=\left(\boldsymbol{Y}^{(l_1)} \otimes_{l_3} \boldsymbol{Y}^{(l_2)}\right)} \,, \tag{7}$$

where the entry of order m_3 for the coupled irreps $\tilde{\boldsymbol{Y}}^{(l_3)}$ is given by

$$\tilde{Y}^{l_3}_{m_3} = \sum_{m_1=-l_1}^{l_1} \sum_{m_2=-l_2}^{l_2} C^{l_1,l_2,l_3}_{m_1,m_2,m_3} Y^{l_1}_{m_1} Y^{l_2}_{m_2} \,, \tag{8}$$

and $C^{l_1,l_2,l_3}_{m_1,m_2,m_3}$ are the so-called *Clebsch-Gordon coefficients*. In the following, we will denote the tensor product of degrees l_1 and l_2 followed by "contraction" to l_3 (meaning the irreps of degree l_3 in the direct sum representation of their tensor product) as $\left(\boldsymbol{Y}^{(l_1)} \otimes_{l_3} \boldsymbol{Y}^{(l_2)}\right)$, which is a mapping of the form $\mathbb{R}^{(2l_1+1)\times(2l_2+1)} \mapsto \mathbb{R}^{2l_3+1}$, since $m_3 \in \{-l_3, \dots l_3\}$.

3 Methods

In the following, we describe the main methodological contributions of this work. We introduce the concept of an adapted point cloud $\mathcal{P}_{3D}(\mathcal{R}, \mathcal{X}, \mathcal{F})$, which incorporates the set of spherical harmonics coordinates (SPHCs) $\mathcal{X} = (\boldsymbol{\chi}_1, \dots, \boldsymbol{\chi}_n)$ (see below) in addition to features $\mathcal{F}$ and Euclidean coordinates $\mathcal{R}$. However, contrary to $\mathcal{R}$, SPHCs $\mathcal{X}$ are refined during the message passing updates. Having SPHCs as part of the molecular point cloud extends the idea of current MPNNs, which learn message functions on $\mathcal{R}$, only. Instead, we learn a message function m (cf. eq. (1)) on both, the (fixed) atomic coordinates $\mathcal{R}$ as well as on the SPHCs $\mathcal{X}$. This adapted message-passing scheme allows to learn non-local geometric corrections. Based on these design principles, we propose the SO3KRATES architecture.

Initialization Feature vectors are initialized from the atomic numbers $z_i \in \mathbb{N}$ (denoting which chemical element an atom belongs to) by an embedding map

$$\mathbf{f}_i = f_{\text{emb}}(z_i), \tag{9}$$

where $f_{\text{emb}} : \mathbb{N} \mapsto \mathbb{R}^F$. We define SPHCs $\boldsymbol{\chi}$ as the concatenation of degrees $\mathcal{L} \coloneqq \{l_{\min}, \dots, l_{\max}\}$

$$\boldsymbol{\chi} = [\ \underbrace{\boldsymbol{\chi}^{(l_{\min})}}_{\in \mathbb{R}^{2l_{\min}+1}}, \dots, \underbrace{\boldsymbol{\chi}^{(l_{\max})}}_{\in \mathbb{R}^{2l_{\max}+1}}\ ] \in \mathbb{R}^{(l_{\max}-l_{\min}+1)^2}, \tag{10}$$

such that their transformation under rotation can be expressed in terms of concatenated Wigner-D matrices (see appendix A.1). The short-hand $\boldsymbol{\chi}^{(l)} \in \mathbb{R}^{2l+1}$ refers to the subset of SPHCs with degree l. They are initialized as

$$\boldsymbol{\chi}_i^{(l)} = \frac{1}{\mathcal{C}_i} \sum_{j \in \mathcal{N}(i)} \phi_{r_{\text{cut}}}(r_{ij}) \cdot \boldsymbol{Y}^{(l)}(\hat{\boldsymbol{r}}_{ij}), \tag{11}$$

where $\mathcal{C}_i = \sum_{j \in \mathcal{N}(i)} \phi_{r_{\text{cut}}}(r_{ij})$, $\phi_{r_{\text{cut}}} : \mathbb{R} \mapsto \mathbb{R}$ is the cosine cutoff function [6], and the sum runs over the neighborhood $\mathcal{N}(i)$ of atom i.

Message Passing Update Two branches of attention-weighted MP steps are defined for the feature vectors $\mathbf{f}$ and SPHCs $\boldsymbol{\chi}$ (see Fig. 1a). After initialization (eqs. (9) and (11)), the features are updated as

$$\mathbf{f}_i' = \mathbf{f}_i + \sum_{j \in \mathcal{N}(i)} \phi_{r_{\text{cut}}}(r_{ij}) \cdot \alpha_{ij} \cdot \mathbf{f}_j\,, \tag{12}$$

where $\alpha_{ij} \in \mathbb{R}$ are self-attention [49, 50] coefficients (see below). In analogy to the feature vectors, it is possible to define an MP update for the SPHCs as

$$\boldsymbol{\chi}_i'^{(l)} = \boldsymbol{\chi}_i^{(l)} + \sum_{j \in \mathcal{N}(i)} \phi_{r_{\text{cut}}}(r_{ij}) \cdot \alpha_{ij}^{(l)} \cdot \boldsymbol{Y}^{(l)}(\hat{\boldsymbol{r}}_{ij})\,, \tag{13}$$

where individual attention coefficients $\alpha_{ij}^{(l)} \in \mathbb{R}$ for each degree of the SPHCs are computed using multi-head attention [49]. However, with this definition, both MP updates are limited to local neighborhoods $\mathcal{N}(i)$. To be able to model non-local effects, we introduce the SPHC distance matrix $\mathbf{X} \in \mathbb{R}^{n \times n}$ with entries $\chi_{ij} = \|\boldsymbol{\chi}_i - \boldsymbol{\chi}_j\|_2$, i.e. distances between two atoms i and j in SPHC space for all possible pair-wise combinations of n atoms. To have uniform scales, we further apply the $\mathrm{softmax}$ along each row of $\mathbf{X}$ to generate a rescaled matrix $\tilde{\mathbf{X}} = \mathrm{softmax}(\mathbf{X})$ with entries $\tilde{\chi}_{ij}$. A polynomial cutoff function $\phi_{\tilde{\chi}_{\text{cut}}}$ [29] is then applied to $\tilde{\mathbf{X}}$ to define spherical neighborhoods $\mathcal{N}_{\boldsymbol{\chi}}(i)$ (see A.2), which may include atoms that are far away in Euclidean space (see Fig. 1f). The spherical cutoff distance is chosen as $\tilde{\chi}_{\text{cut}} = 1/n$ to ensure that spherical neighborhoods remain small, even

when going to larger molecules. We then incorporate non-local geometric corrections into the MP update of the SPHCs as

$$\boldsymbol{\chi}_i'^{(l)} = \boldsymbol{\chi}_i^{(l)} + \underbrace{\sum_{j\in\mathcal{N}(i)} \phi_{r_{\text{cut}}}(r_{ij}) \cdot \alpha_{ij}^{(l)} \cdot \boldsymbol{Y}^{(l)}(\hat{\boldsymbol{r}}_{ij})}_{\text{local in } \mathbb{R}^3} + \underbrace{\sum_{j\in\mathcal{N}_{\boldsymbol{\chi}}(i)} \phi_{\tilde{\chi}_{\text{cut}}}(\tilde{\chi}_{ij}) \cdot \alpha_{ij}^{(l)} \cdot \boldsymbol{Y}^{(l)}(\hat{\boldsymbol{r}}_{ij})}_{\text{local in } \boldsymbol{\chi}\text{, but non-local in } \mathbb{R}^3} . \quad (14)$$

We will show in the first part of the experiments, how geometric corrections from SPHC space allow for modelling non-local quantum effects, inaccessible to current architectures. In the second part, we use a SO3KRATES model without geometric corrections, which makes it a traditional MPNN in the sense of only localizing in $\mathbb{R}^3$. We find this architecture to be highly parameter, data and time efficient while capable of reaching SOTA results.

Spherical Filter and Self-Attention The self-attention coefficients in eqs. (12)–(14) are calculated as

$$\alpha_{ij} = \mathbf{f}_i^T(\mathbf{w}_{ij} \odot \mathbf{f}_j)/\sqrt{F}\,, \quad (15)$$

where $\mathbf{w}_{ij} \in \mathbb{R}^F$ is the output of a filter generating function and '$\odot$' denotes the element-wise product. The filter maps the Euclidean distance r_{ij} and per-degree SPHC distances $\chi_{ij}^{(l)} = \|\boldsymbol{\chi}_j^{(l)} - \boldsymbol{\chi}_i^{(l)}\|_2$ between the current SPHCs of atoms i and j into the feature space $\mathbb{R}^F$ (as a short-hand, we write the vector containing all per-degree SPHC distances as $[\chi_{ij}^{(l)}]_{l\in\mathcal{L}}$). It is built as the linear combination of two filter-generating functions

$$\mathbf{w}_{ij} = \underbrace{\phi_r(r_{ij})}_{\text{radial filter}} + \underbrace{\phi_s\left([\chi_{ij}^{(l)}]_{l\in\mathcal{L}}\right)}_{\text{spherical filter}}, \quad (16)$$

which separately act on the Euclidean and SPHC distances. We call $\phi_r : \mathbb{R} \mapsto \mathbb{R}^F$ the *radial filter* function and $\phi_s : \mathbb{R}^{|\mathcal{L}|} \mapsto \mathbb{R}^F$ the *spherical filter* function (an ablation study for ϕ_s can be found in appendix A.4). Since per-atom features $\mathbf{f}_i$, interatomic distances r_{ij}, and per-degree distances $\chi_{ij}^{(l)}$ are invariant under rotations (proof in appendix A.1), so are the self-attention coefficients α_{ij}.

While we choose to pass the per-degree norms directly into the filter generating function ϕ_s, future work might explore the possibilities of alternative metrics (instead of the L2 norm) or an expansion in terms of basis functions as it is common practice for inter-atomic distances (see appendix A.3 eq. (30)).

Atomwise Interaction After each MP update, features and SPHCs are coupled with each other according to

$$\mathbf{f}_i' = \mathbf{f}_i + \phi_1\left(\mathbf{f}_i, [\chi_i^{(l)}]_{l\in\mathcal{L}}, [\tilde{\chi}_i^{(l)}]_{l\in\mathcal{L}}\right)\,, \quad (17)$$

$$\boldsymbol{\chi}_i'^{(l)} = \boldsymbol{\chi}_i^{(l)} + \phi_2^{(l)}\left(\mathbf{f}_i, [\chi_i^{(l)}]_{l\in\mathcal{L}}, [\tilde{\chi}_i^{(l)}]_{l\in\mathcal{L}}\right)\boldsymbol{\chi}_i^{(l)} + \phi_3^{(l)}\left([\tilde{\chi}_i^{(l)}]_{l\in\mathcal{L}}\right)\tilde{\boldsymbol{\chi}}_i^{(l)}\,, \quad (18)$$

where $\phi_1 : \mathbb{R}^{F+2|\mathcal{L}|} \mapsto \mathbb{R}^F$, $\phi_2^{(l)} : \mathbb{R}^{F+2|\mathcal{L}|} \mapsto \mathbb{R}$, and $\phi_3^{(l)} : \mathbb{R}^{|\mathcal{L}|} \mapsto \mathbb{R}$. In the inputs to $\phi_{1,2,3}$, degree-wise scalars $\chi^{(l)} = \|\boldsymbol{\chi}^{(l)}\|_2$ are used to preserve equivariance. The coupling step additionally includes cross-degree coupled SPHCs $\tilde{\boldsymbol{\chi}}_i^{(l)}$ for each degree l. Following [48] they are constructed as

$$\tilde{\boldsymbol{\chi}}_i^{(l)} = \sum_{l_1=l_{\min}}^{l_{\max}} \sum_{l_2=l_1+1}^{l_{\max}} k_{l_1,l_2,l}\left(\boldsymbol{\chi}_i^{(l_1)} \otimes_l \boldsymbol{\chi}_i^{(l_2)}\right), \quad (19)$$

where $k_{l_1,l_2,l} \in \mathbb{R}$ are learnable coefficients for all valid combinations of l_1, l_2 given l and the term in brackets is the contraction of degrees l_1 and l_2 into degree l (eq. (8)).

SO3KRATES architecture Using the design paradigm above, we build the transformer network SO3KRATES, which consists of a self-attention block on $\mathcal{F}$ and $\mathcal{X}$ (eqs. (12) and (13)), respectively, as well as an interaction block (eqs. (17) and (18)) per layer. After initialization of the features and the SPHCs according to eqs. (9) and (11), they are updated iteratively by passing through n_l layers. Atomic energy contributions $E_i \in \mathbb{R}$ are predicted from the features of the final layer using a two-layered output block. The individual contributions are summed to the total energy prediction $E = \sum_i^n E_i$. See Fig. 1b for an overview. More details on the implementation, training details and network hyperparameters are given in appendix A.3 and A.13.

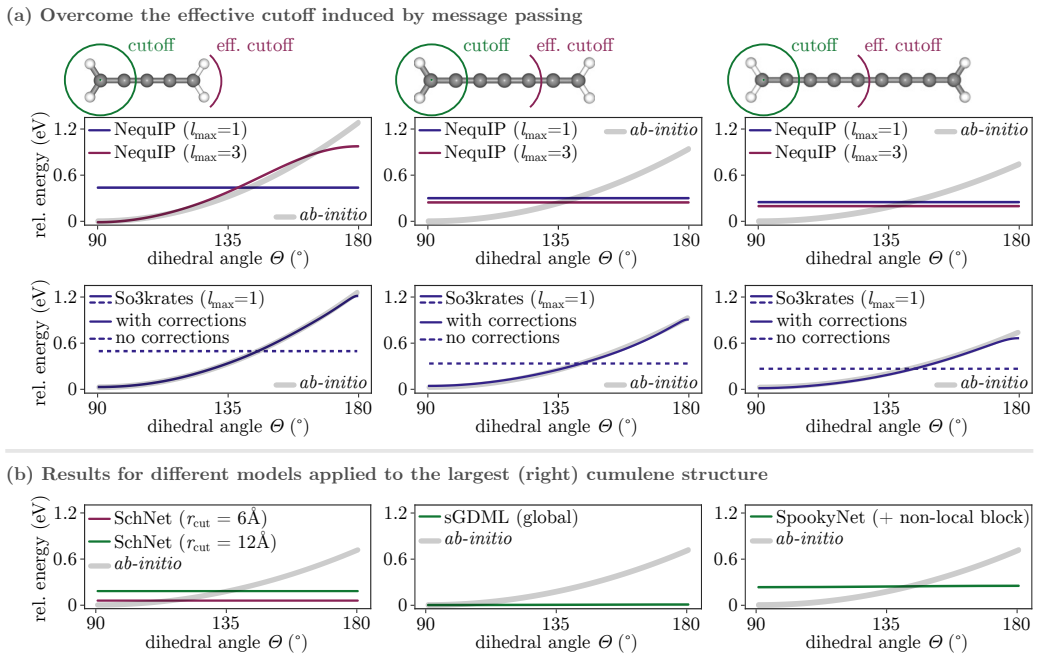

Figure 2: **(a)** Energy predictions for different cumulene structures with NEQUIP ($l_{\max} = 1$ and $l_{\max} = 3$) and SO3KRATES ($l_{\max} = 1$) models with and without geometric corrections (eqs. (13) and (14)). Cutoff radius and number of layers are kept constant at $r_{\text{cut}} = 2.5\,\text{Å}$ and $n_l = 4$, respectively. **(b)** We additionally test other SOTA models on the largest cumulene structure (9 carbons). We find, that even for a very large cutoff, SCHNET fails to describe the dihedral angle profile. The same holds true for an inherently global sGDML model and even for a SPOOKYNET model that explicitly includes non-local corrections.

4 Experiments

In the first subsection, we show how non-local quantum effects can be incorporated by using non-local corrections from the space of SPHCs. In the second part of the experiments, we remove the non-local part which yields a traditional, $\mathbb{R}^3$-local MPNN which reaches SOTA results on established benchmarks while requiring much less computational time and parameters than competitive models. A scaling analysis as well as an accuracy comparison for both model variants can be found in appendix A.8 and A.5.

Non-Local Geometric Interactions For efficiency reasons, MPNNs only consider interactions between atoms in local neighborhoods, i.e. within a cutoff radius r_{cut}. Thus, information can only be propagated over a distance of r_{cut} within a single MP step. Although multiple MP updates increase the effective cutoff distance, because information can "hop" between different neighborhoods as long as they share at least one atom, each MP step is accompanied by an undesirable loss of information, which limits the accuracy that can be obtained. Consequently, MPNNs are unable to describe non-local effects on length-scales that exceed the effective cutoff distance. To illustrate this problem, we consider the challenging open task [21] of learning the potential energy of cumulene molecules with different sizes (see Fig. 2a). Here, the relative orientation of the hydrogen rotors at the far ends of the molecule strongly influences its energy due to non-local electronic effects [21]. In order to be able to successfully learn the energy profile with a local model, the effective cutoff has to be large enough to allow information to propagate from one hydrogen rotor to the other.

As a representative example for MPNNs, we consider the recently proposed NEQUIP model [32], which achieves SOTA performance on several benchmarks. We find that even when the effective cutoff radius is large enough in principle, an MPNN with $n_l = 4$, $r_{\text{cut}} = 2.5\,\text{Å}$, and $l_{\max} \leq 1$ fails to learn the correct energy profile. This is due to the fact that the relevant geometric information "cancels out" (similar to addition of vectors oriented in opposite directions) within each neighborhood,

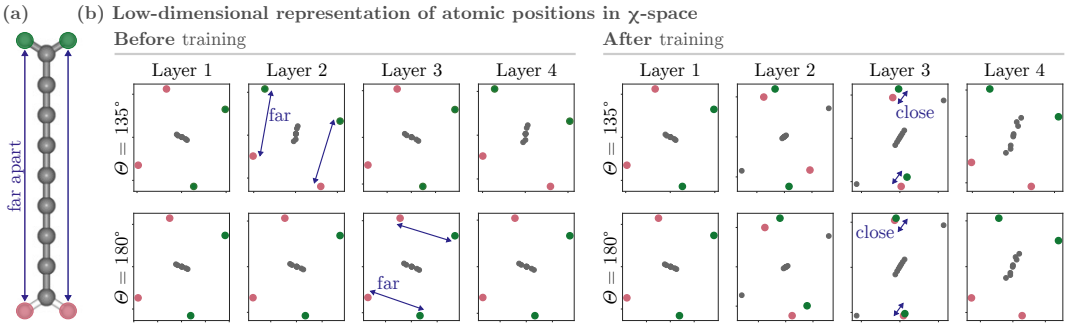

Figure 3: **(a)** Cumulene structure in Euclidean space. **(b)** Low-dimensional projections of the SPHCs ($l_{\max} = 1$) in SO3KRATES before and after training. Carbon atoms are grey and hydrogen atoms on different ends of the molecule are green and red, respectively. During training, the model maps hydrogen atoms at opposite ends close together in SPHC space (which is not the case at initialization).

underlining the limited expressiveness of mean-field interactions in MPNNs. Only by including higher-order geometric correlations, e.g. going to $l_{\max} = 3$, the correct energy profile can be recovered (at the cost of computational efficiency). When going to even larger cumulene structures, however, the effective cutoff becomes too small and it is necessary to increase the number of MP layers to solve the task (again, at the cost of lower computational efficiency), which is illustrated in appendix A.11. Neither increasing the maximum degree of interactions $l_{\max}$, nor the number of layers n_l, is a satisfactory workaround: Instead of offering a general solution to describe non-local interactions, both options decrease computational efficiency, while only shifting the problem to larger length-scales or higher-order geometric correlations.

We further apply three additional models to the cumulene structure with nine carbon atoms. To that end, we use an invariant SCHNET model with varying cutoff distances (6 Å and 12 Å), an inherently global but invariant sGDML model and the SPOOKYNET architecture which explicitly includes global effects using a non-local block. We find that none of the three is capable of describing the rotor energy profile of cumulene.

In contrast, our proposed SO3KRATES architecture is able to reproduce the energy profile for cumulene molecules of all sizes independent of the effective cutoff radius. Crucially, even with $l_{\max} = 1$, the predicted energy matches the *ab-initio* reference faithfully. We find that geometric corrections in the MP update of the SPHCs (cf. eq. (14)) are responsible for the increased capability of describing higher-order geometric correlations, as a SO3KRATES model with a naive MP update (cf. eq. (13)) fails to solve this task with $l_{\max} = 1$ (see Fig. 2a). We further confirm that the model picks up on the physically relevant interaction between the hydrogen rotors by analysing the attention values after training (see Fig. 8 appendix A.7). To illustrate how SO3KRATES is able to describe non-local effects, we show a low-dimensional projection of the atomic SPHCs before and after training for the largest of the cumulene molecules (Fig. 3). After training, the SPHCs for hydrogen atoms at opposite ends of the molecule are embedded close together in SPHC space, allowing SO3KRATES to efficiently model the non-local geometric dependence between the hydrogen rotors.

Generalization to structures, larger than those in the training data are usually associated with the re-usability of the learned, local representations. For that reason, it is unclear if this property still holds when non-local corrections are used. As we show in appendix A.5 a SO3KRATES model with non-local corrections still generalizes well to completely unknown and larger structures.

Benchmarks, Data Efficiency and Generalization As pointed out in [31] and [32], equivariant features not only increase performance, but also improve data efficiency. The latter is particularly important, as *ab-initio* methods for reference data generation can become exceedingly expensive when high accuracy is required. Here, we use a subset of the recently introduced QM7-X data set [51], which we call QM7-X250. It contains 250 different molecular structures, each with 80 data points for training, 10 data points for validation and 11-3748 data points for testing (for details, see appendix A.9). The small number of training/validation samples per molecule makes it particularly suited for evaluating model behavior in the low data regime. In the following, we train (1) one model

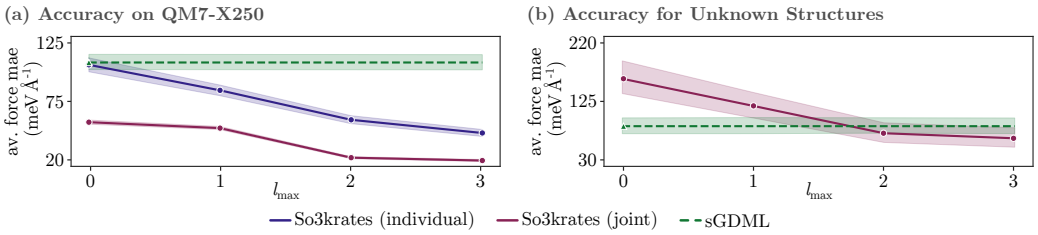

Figure 4: **(a)** Average force MAE as a function of $l_{\max}$ on 250 structures from QM7-X [51]. The blue solid line corresponds to SO3KRATES models trained on each of the 250 structures (à 80 training points), individually. The jointly trained SO3KRATES model (red solid line) is a single model trained on all 250×80 training points. **(b)** Accuracy of force predictions for unknown structures as function of $l_{\max}$. Generalization is only investigated for the jointly trained model. Due to the way the molecular descriptor is designed, sGDML [38] models can only be trained on individual structures.

per structure in QM7-X250 and (2) one model for all structures in QM7-X250 ($250 \times 80 = 20$k training points), which we refer to as *individual* and *joint* models, respectively.

We start by investigating the performance as a function of the maximal degree $l_{\max}$ and find that the error strongly decreases with higher $l_{\max}$ (Fig. 4a). As kernel methods are known to perform well in the low-data regime [21], we compare our results to sGDML [38] kernel models, which only use distances as a molecular descriptor (corresponding to $l_{\max} = 0$). For $l_{\max} = 0$, we find sGDML gives competitive results, whereas for $l_{\max} = 1$, SO3KRATES starts to outperform sGDML. As soon as $l_{\max} \geq 2$, however, the prediction accuracy of SO3KRATES is greater than that of sGDML by a large margin. Thus, increasing the order of geometric information in the SPHCs leads to strong improvements in the low-data regime. For jointly trained models, we find that SO3KRATES outperforms sGDML even for $l_{\max} = 0$, with continuous improvement for increasing $l_{\max}$. In appendix A.10 we report energy *and* force errors across degrees and further experimental details.

The generalization capability of SO3KRATES is tested, by applying a jointly trained model to 25 completely unknown molecules from the QM7-X data set (see, Fig. 4b, details in appendix A.10). Again, we find that force MAEs decrease with increasing $l_{\max}$. For reference, we compare SO3KRATES to individually trained sGDML models and find that SO3KRATES performs on par, or even slightly better, for $l_{\max} \geq 2$. Going beyond $l_{\max} = 2$ is found to only marginally improve generalization. In addition, we report results for a model trained on the full QM7-X data set in appendix A.5, following [30].

For completeness, we also apply SO3KRATES to the popular MD17 benchmark (see Table 1). We find, that SO3KRATES outperforms networks that have the same parameter complexity by a large margin (PAINN and NEWTONNET). Notably, it requires significantly less parameters than other SH based architectures (NEQUIP and SPOOKYNET), while performing only slightly worse or even on par with them. Furthermore, SO3KRATES outperforms DIMENET, its closest competitor in timing (cf. Fig. 1.d), consistently by a large margin. Compared to current SH based approaches, GEMNETQ needs less parameters (still ~ 2.5x more than SO3KRATES) to achieve competitive results. However, it requires the explicit calculation of dihedral angles which scales cubically in the number of neighboring atoms. Due to its linear scaling (see A.8) and lightweight structure, SO3KRATES can significantly reduce the time for training and inference (see Fig. 1d and A.6).

5 Discussion and Conclusion

Due to the locality assumption used in most MPNNs, they are unable to model non-local electronic effects, which result in global geometric dependencies between different parts of a molecule. The length-scales of such interactions often greatly exceed the cutoff radius used in the MP step, and even though stacking multiple MP layers increases the effective cutoff, ultimately, MPNNs are not capable of efficiently modeling geometric dependencies on arbitrary length scales.

In this work, we contribute conceptually by proposing an efficient and scalable solution to this problem. We suggest a set of refinable, equivariant coordinates for point clouds in Euclidean space, called spherical harmonic coordinates (SPHCs). Non-local geometric effects can then be efficiently

Table 1: MAE for energy (in kcal mol^{-1}) and forces (in kcal mol^{-1} Å^{-1}) of current state-of-the-art machine learning models on the MD17 benchmark for 1k training points. For each model, the number of paramaeters, as well as the scaling in the number of neighbors m is shown.

		NEQUIP [32] 3M | $\mathcal{O}(m)$	SPOOKYNET [30] 3M | $\mathcal{O}(m)$	GEMNETQ [34] 2.2M | $\mathcal{O}(m^3)$	DIMENET [29] 1.9M | $\mathcal{O}(m^2)$	PAINN [31] 600k | $\mathcal{O}(m)$	NEWTONNET [52] 500k | $\mathcal{O}(m)$	SO3KRATES 700k | $\mathcal{O}(m)$
Aspirin	*energy*	0.13	0.151	–	0.204	0.159	0.168	0.139
	forces	0.19	0.258	0.217	0.499	0.371	0.348	0.236
Ethanol	*energy*	0.05	0.052	–	0.064	0.063	0.061	0.052
	forces	0.09	0.094	0.088	0.230	0.230	0.211	0.096
Malondialdehyde	*energy*	0.08	0.079	–	0.104	0.091	0.096	0.077
	forces	0.13	0.167	0.159	0.383	0.319	0.323	0.147
Naphthalene	*energy*	0.11	0.116	–	0.122	0.117	0.118	0.115
	forces	0.04	0.089	0.051	0.215	0.083	0.084	0.074
Salicyclic Acid	*energy*	0.11	0.114	–	0.134	0.114	0.115	0.106
	forces	0.09	0.180	0.124	0.374	0.209	0.197	0.145
Toluene	*energy*	0.09	0.094	–	0.102	0.097	0.094	0.095
	forces	0.05	0.087	0.060	0.216	0.102	0.088	0.073
Uracil	*energy*	0.10	0.105	–	0.115	0.104	0.107	0.103
	forces	0.08	0.119	0.104	0.301	0.140	0.149	0.111

modeled by including geometric corrections, which are localized in the space of SPHCs, but non-local in Euclidean space. Further, we show that introducing spherical filter functions acting on the SPHCs increases geometric resolution and predictive accuracy.

We then propose the SO3KRATES architecture, a self-attention based MPNN, which decouples atomic features from higher-order geometric information. This allows to drastically decrease the parametric complexity while still achieving SOTA prediction accuracy. We show evidence that increasing the geometric order of SPHCs greatly improves model performance in the low-data regime, as well as generalization to unknown molecules.

A limitation of the current implementation of SO3KRATES is that spherical neighborhoods $\mathcal{N}_{\boldsymbol{x}}$ in eq. (14) are computed from all pairwise distances in SPHC space. An alternative implementation could use a space partitioning scheme to find neighborhoods more efficiently. In a broader context, our work falls into the category of approaches that can help to reduce the vast computational complexity of molecular and material simulations. This can accelerate novel drug and material designs, which holds the promise of tackling societal challenges, such as climate change and sustainable energy supply [53]. Of course, our method could also be used for nefarious applications, e.g. design of chemical warfare, but this is true for all quantum chemistry methods.

Future research will focus on applications of SO3KRATES to materials and bio-molecules, which are typical examples of chemical systems where the accurate description of non-local effects is necessary to produce novel insights. Efficient treatment of non-local effects in point cloud data goes beyond the domain of quantum chemistry. One way of representing non-local dependencies are non-local neural networks [54]. In comparison to the presented approach they compute a relation in feature rather than in Euclidean space, making it incapable of capturing direct geometric relations in Euclidean space. However, this might be necessary if the relative orientation of objects far apart from each other plays a role for identifying different objects.

6 Acknowledgements

All authors acknowledge support by the Federal Ministry of Education and Research (BMBF) for BIFOLD (01IS18037A). KRM was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government(MSIT) (No. 2019-0-00079, Artificial Intelligence Graduate School Program, Korea University and No. 2022-0-00984, Development of Artificial Intelligence Technology for Personalized Plug-and-Play Explanation and Verification of Explanation), and was partly supported by the German Ministry for Education and Research (BMBF) under Grants 01IS14013A-E, AIMM, 01GQ1115, 01GQ0850, 01IS18025A and 01IS18037A; the German Research Foundation (DFG). We thank Stefan Chmiela, Mihail Bogojeski and Nicklas Schmitz for helpful discussions and feedback on the manuscript.

References

[1] Mark E Tuckerman. Ab initio molecular dynamics: basic concepts, current trends and novel applications. *J. Phys. Condens. Matter*, 14(50):R1297, 2002.

[2] Frank Noé, Alexandre Tkatchenko, Klaus-Robert Müller, and Cecilia Clementi. Machine learning for molecular simulation. *Annu. Rev. Phys. Chem.*, 71:361–390, 2020.

[3] John A Keith, Valentin Vassilev-Galindo, Bingqing Cheng, Stefan Chmiela, Michael Gastegger, Klaus-Robert Müller, and Alexandre Tkatchenko. Combining machine learning and computational chemistry for predictive insights into chemical systems. *Chemical Reviews*, 121 (16):9816–9872, 2021. URL `https://pubs.acs.org/doi/abs/10.1021/acs.chemrev.1c00107`.

[4] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.*, 98(14):146401, 2007.

[5] Albert P Bartók, Mike C Payne, Risi Kondor, and Gábor Csányi. Gaussian Approximation Potentials: the accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.*, 104(13): 136403, 2010.

[6] Jörg Behler. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *J. Chem. Phys.*, 134(7):074106, 2011.

[7] Albert P Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Phys. Rev. B*, 87(18):184115, 2013.

[8] Zhenwei Li, James R Kermode, and Alessandro De Vita. Molecular dynamics with on-the-fly machine learning of quantum-mechanical forces. *Phys. Rev. Lett.*, 114(9):096405, 2015.

[9] 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. *Sci. Adv.*, 3(5):e1603015, 2017.

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

[11] Michael Gastegger, Jörg Behler, and Philipp Marquetand. Machine learning molecular dynamics for the simulation of infrared spectra. *Chem. Sci.*, 8(10):6924–6935, 2017.

[12] Stefan Chmiela, Huziel E. Sauceda, Klaus-Robert Müller, and Alexandre Tkatchenko. Towards exact molecular dynamics simulations with machine-learned force fields. *Nat. Commun.*, 9(1): 3887, 2018. doi: 10.1038/s41467-018-06169-2.

[13] Kristof T Schütt, Huziel E Sauceda, Pieter-Jan Kindermans, Alexandre Tkatchenko, and Klaus-Robert Müller. SchNet – a deep learning architecture for molecules and materials. *J. Chem. Phys.*, 148(24):241722, 2018.

[14] 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.

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

[16] Nicholas Lubbers, Justin S Smith, and Kipton Barros. Hierarchical modeling of molecular energies using a deep neural network. *J. Chem. Phys.*, 148(24):241715, 2018.

[17] Martin Stöhr, Leonardo Medrano Sandonas, and Alexandre Tkatchenko. Accurate many-body repulsive potentials for density-functional tight-binding from deep tensor neural networks. *J. Phys. Chem. Lett.*, 11:6835–6843, 2020.

[18] Felix A Faber, Anders S Christensen, Bing Huang, and O Anatole von Lilienfeld. Alchemical and structural distribution based representation for universal quantum machine learning. *J. Chem. Phys.*, 148(24):241717, 2018.

[19] 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.

[20] Anders S Christensen, Lars A Bratholm, Felix A Faber, and O Anatole von Lilienfeld. FCHL revisited: faster and more accurate quantum machine learning. *J. Chem. Phys.*, 152(4):044107, 2020.

[21] 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*, 121(16):10142–10186, 2021.

[22] Yaolong Zhang, Ce Hu, and Bin Jiang. Embedded Atom Neural Network Potentials: efficient and accurate machine learning with a physically inspired representation. *J. Phys. Chem. Lett.*, 10(17):4962–4967, 2019.

[23] Silvan Käser, Oliver Unke, and Markus Meuwly. Reactive dynamics and spectroscopy of hydrogen transfer from neural network-based reactive potential energy surfaces. *New J. Phys.*, 22:55002, 2020.

[24] O Anatole von Lilienfeld, Klaus-Robert Müller, and Alexandre Tkatchenko. Exploring chemical compound space with quantum-based machine learning. *Nat. Rev. Chem.*, 4(7):347–358, 2020.

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

[26] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[27] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[28] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[29] Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123*, 2020.

[30] 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. *Nat. Commun.*, 12:7273, 2021.

[31] 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.

[32] 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. *arXiv preprint arXiv:2101.03164*, 2021.

[33] 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.

[34] Johannes Klicpera, Florian Becker, and Stephan Günnemann. Gemnet: Universal directional graph neural networks for molecules. *arXiv preprint arXiv:2106.08903*, 2021.

[35] Sergey N Pozdnyakov, Michael J Willatt, Albert P Bartók, Christoph Ortner, Gábor Csányi, and Michele Ceriotti. Incompleteness of atomic structure representations. *Physical Review Letters*, 125(16):166001, 2020.

[36] Fabian B Fuchs, Daniel E Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *arXiv preprint arXiv:2006.10503*, 2020.

[37] Aneja Alisha. Performance comparison between nvidia's geforce gtx 1080 and tesla p100 for deep learning. `https://github.com/alisha17/benchmarks`, 2017.

[38] Stefan Chmiela, Huziel E Sauceda, Igor Poltavsky, Klaus-Robert Müller, and Alexandre Tkatchenko. sgdml: Constructing accurate and data efficient molecular force fields using machine learning. *Computer Physics Communications*, 240:38–45, 2019.

[39] Brandon Anderson, Truong-Son Hy, and Risi Kondor. Cormorant: covariant molecular neural networks. *arXiv preprint arXiv:1906.04015*, 2019.

[40] Philipp Thölke and Gianni De Fabritiis. Equivariant transformers for neural network based molecular potentials. In *International Conference on Learning Representations*, 2021.

[41] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *arXiv preprint arXiv:1803.10091*, 2018.

[42] J-F Cardoso and Beate H Laheld. Equivariant adaptive source separation. *IEEE Transactions on signal processing*, 44(12):3017–3030, 1996.

[43] Antonios K Alexandridis and Achilleas D Zapranis. *Wavelet neural networks: with applications in financial engineering, chaos, and classification*. John Wiley & Sons, 2014.

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

[45] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International conference on artificial neural networks*, pages 44–51. Springer, 2011.

[46] Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: exact likelihood generative learning for symmetric densities. In *International Conference on Machine Learning*, pages 5361–5370. PMLR, 2020.

[47] Eugene Paul Wigner. Gruppentheorie und ihre anwendung auf die quantenmechanik der atomspektren. 1931.

[48] 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, 2021.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[50] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[51] Johannes Hoja, Leonardo Medrano Sandonas, Brian G Ernst, Alvaro Vazquez-Mayagoitia, Robert A DiStasio Jr, and Alexandre Tkatchenko. Qm7-x, a comprehensive dataset of quantum-mechanical properties spanning the chemical space of small organic molecules. *Scientific data*, 8(1):1–11, 2021.

[52] Mojtaba Haghighatlari, Jie Li, Xingyi Guan, Oufan Zhang, Akshaya Das, Christopher J Stein, Farnaz Heidar-Zadeh, Meili Liu, Martin Head-Gordon, Luke Bertels, et al. Newtonnet: A newtonian message passing network for deep learning of interatomic potentials and forces. *arXiv preprint arXiv:2108.02913*, 2021.

[53] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, et al. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL materials*, 1(1):011002, 2013.

[54] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[55] Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX, 2020. URL `http://github.com/google/flax`.

[56] Matteo Hessel, David Budden, Fabio Viola, Mihaela Rosca, Eren Sezener, and Tom Hennigan. Optax: composable gradient transformation and optimisation, in jax!, 2020. URL `http://github.com/deepmind/optax`.

[57] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL `https://doi.org/10.1038/s41586-020-2649-2`.

[58] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/google/jax`.

[59] Ameen Ali, Thomas Schnake, Oliver Eberle, Grégoire Montavon, Klaus-Robert Müller, and Lior Wolf. Xai for transformers: better explanations through conservative propagation. *arXiv preprint arXiv:2202.07304*, 2022.

[60] Tsz Wai Ko, Jonas A Finkler, Stefan Goedecker, and Jörg Behler. A fourth-generation high-dimensional neural network potential with accurate electrostatics including non-local charge transfer. *Nature communications*, 12(1):1–11, 2021.

[61] Diederik P Kingma and Jimmy Ba. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (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] We did not include theoretical results.
   (b) Did you include complete proofs of all theoretical results? [N/A] Even though we proof equivariance, this is not a theoretical result of the presented work, as equivariance w. r. t. $\mathrm{SO}(3)$ is well understood.
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]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Fig. 4 or Fig. 6.
   (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]

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]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] No such data is used.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] No such data is used.
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] Neither crowdsourcing nor research on human subjects was conducted.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] Neither crowdsourcing nor research on human subjects was conducted.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] Neither crowdsourcing nor research on human subjects was conducted.

A Appendix

A.1 Proof of Equivariance

SPHC Initialization Here we give proof that certain quantities from the main text are invariant or equivariant, respectively. Let us start with the spherical harmonic coordinates (SPHC) which are initialized as

$$\boldsymbol{\chi}_i^{(l)}(R) = \frac{1}{\mathcal{C}_i} \sum_{j \in \mathcal{N}(i)} \phi_{r_{\text{cut}}}(r_{ij}) \cdot \boldsymbol{Y}^{(l)}(\hat{\boldsymbol{r}}_{ij}) \,, \tag{20}$$

where $\boldsymbol{Y}^{(l)} : S^2 \mapsto \mathbb{R}^{2l+1}$. In contrast to the main text, we make the dependence of the right hand side of eq. (20) on the atomic positions $\mathcal{R} = \{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_n\}$ explicit. Each degree-wise entry in the initialized SPHCs (eq. (20)) transforms under rotation as

$$\begin{aligned}
\boldsymbol{\chi}_i^{(l)}(R_\theta\, \mathcal{R}) &\sim \sum_{j \in \mathcal{N}(i)} \phi_{r_{\text{cut}}}(r_{ij}) \cdot \boldsymbol{Y}^{(l)}(R_\theta \hat{\boldsymbol{r}}_{ij}) \\
&= \sum_{j \in \mathcal{N}(i)} \phi_{r_{\text{cut}}}(r_{ij}) \cdot \boldsymbol{D}^l(R_\theta)\, \boldsymbol{Y}^{(l)}(\hat{\boldsymbol{r}}_{ij}) \\
&= \boldsymbol{D}^l(R_\theta) \sum_{j \in \mathcal{N}(i)} \phi_{r_{\text{cut}}}(r_{ij}) \cdot \boldsymbol{Y}^{(l)}(\hat{\boldsymbol{r}}_{ij}) \\
&= \boldsymbol{D}^l(R_\theta)\, \boldsymbol{\chi}_i^{(l)}(\mathcal{R}) \,,
\end{aligned} \tag{21}$$

where $\boldsymbol{D}(R_\theta)^l \in \mathbb{R}^{(2l+1)\times(2l+1)}$ is the Wigner-D matrix for degree l given rotation R_θ and $R_\theta \mathcal{R} = \{R_\theta \boldsymbol{r}_1, \ldots, R_\theta \boldsymbol{r}_n\}$. Since the cutoff function $\phi_{r_{\text{cut}}}$ takes the inter-atomic distance $r_{ij} = \|\boldsymbol{r}_j - \boldsymbol{r}_i\|_2$ as its input argument it is always invariant under rotation. Thus, each degree-wise entry $\boldsymbol{\chi}^{(l)}$ is equivariant after the initialization.

SPHC Message Passing Update After initialization, each per-degree entry in $\boldsymbol{\chi}$ is updated as

$$\boldsymbol{\chi}_i'^{(l)}(\mathcal{R}) = \boldsymbol{\chi}_i^{(l)}(\mathcal{R}) + \sum_{j \in \mathcal{N}(i)} \phi_{r_{\text{cut}}}(r_{ij}) \cdot \alpha_{ij}^{(l)} \cdot \boldsymbol{Y}^{(l)}(\hat{\boldsymbol{r}}_{ij}) \,, \tag{22}$$

where $\alpha_{ij}^{(l)}$ are rotationally invariant, per-degree self-attention coefficients. In the first layer, the first part in eq. (22) corresponds to the initialized SPHC entries, which have been already shown to be equivariant (see above). The second part of the equation has the same structural form as the initialization, with the additional self-attention value, which is a rotationally invariant scalar. Thus we can write

$$\begin{aligned}
\boldsymbol{\chi}_i'^{(l)}(R_\theta\, \mathcal{R}) &= \boldsymbol{D}^l(R_\theta)\, \boldsymbol{\chi}^{(l)}(\mathcal{R}) + \sum_{j \in \mathcal{N}(i)} \phi_{r_{\text{cut}}}(r_{ij}) \cdot \alpha_{ij}^{(l)} \cdot \boldsymbol{Y}^{(l)}(R_\theta \hat{\boldsymbol{r}}_{ij}) \\
&= \boldsymbol{D}^l(R_\theta)\, \boldsymbol{\chi}^{(l)}(\mathcal{R}) + \boldsymbol{D}^l(R_\theta) \sum_{j \in \mathcal{N}(i)} \phi_{r_{\text{cut}}}(r_{ij}) \cdot \alpha_{ij}^{(l)} \cdot \boldsymbol{Y}^{(l)}(\hat{\boldsymbol{r}}_{ij}) \\
&= \boldsymbol{D}^l(R_\theta) \left(\boldsymbol{\chi}_i^{(l)}(\mathcal{R}) + \sum_{j \in \mathcal{N}(i)} \phi_{r_{\text{cut}}}(r_{ij}) \cdot \alpha_{ij}^{(l)} \cdot \boldsymbol{Y}^{(l)}(\hat{\boldsymbol{r}}_{ij}) \right) \\
&= \boldsymbol{D}^l(R_\theta)\, \boldsymbol{\chi}_i'^{(l)}(\mathcal{R}) \,,
\end{aligned} \tag{23}$$

which shows that the updated $\boldsymbol{\chi}'^{(l)}$ is also equivariant. The proof of equivariance for later layers follows analogously.

Transformation of the SPHCs After having shown that each per-degree entry of $\boldsymbol{\chi}$ transforms under rotation according to the corresponding Wigner-D matrix $\mathbf{D}^l(R_\theta) \in \mathbb{R}^{(2l+1)\times(2l+1)}$, one can write the direct sum of all Wigner-D matrices as concatenation of matrices along the diagonal

$$\boldsymbol{D}(R_\theta) = \bigoplus_{l \in \mathcal{L}} \boldsymbol{D}^l(R_\theta) \,, \tag{24}$$

such that $\boldsymbol{D}(R_\theta) \in \mathbb{R}^{(l_{\max}-l_{\min}+1)\times(l_{\max}-l_{\min}+1)}$ has a block diagonal structure. The, full SPHC vectors transform under rotation as

$$\boldsymbol{D}(R_\theta)\,\boldsymbol{\chi}(\mathcal{R}) = [\boldsymbol{D}^{l_{\min}}(R_\theta)\boldsymbol{\chi}^{(l_{\min})}(\mathcal{R}), \ldots, \boldsymbol{D}^{l_{\max}}(R_\theta)\boldsymbol{\chi}^{(l_{\max})}(\mathcal{R})]\,. \tag{25}$$

As each of the blocks along the diagonal of $\boldsymbol{D}(R_\theta)$ is an orthogonal Wigner-D matrix, $\boldsymbol{D}(R_\theta)$ itself is also orthogonal.

A.1.1 Invariance of the (per-degree) Norm

The per degree norm is used as input to the spherical filter function ϕ_s. As shown above, each of the per-degree entries in $\boldsymbol{\chi}$ transforms under rotation as

$$\boldsymbol{\chi}^{(l)}(R_\theta\,\mathcal{R}) = \boldsymbol{D}^l(R_\theta)\boldsymbol{\chi}^{(l)}(\mathcal{R})\,. \tag{26}$$

The squared norm can be expressed in terms of an inner product

$$\begin{aligned}
\|\boldsymbol{\chi}^{(l)}(R_\theta\mathcal{R})\|_2^2 &= \left(\boldsymbol{\chi}^{(l)}(R_\theta\,\mathcal{R})\right)^T \boldsymbol{\chi}^{(l)}(R_\theta\,\mathcal{R}) \\
&= (\boldsymbol{D}^l(R_\theta)\,\boldsymbol{\chi}^{(l)}(\mathcal{R}))^T\,(\boldsymbol{D}^l(R_\theta)\,\boldsymbol{\chi}^{(l)}(\mathcal{R})) \\
&= \left(\boldsymbol{\chi}^{(l)}(\mathcal{R})\right)^T \underbrace{\left(\boldsymbol{D}^l(R_\theta)\right)^T \boldsymbol{D}^l(R_\theta)}_{=\mathbb{I}}\,\boldsymbol{\chi}^{(l)}(\mathcal{R}) \\
&= \|\boldsymbol{\chi}^{(l)}(\mathcal{R})\|_2^2,
\end{aligned} \tag{27}$$

where we use the orthogonality of Wigner-D matrices to show that the inner product is rotationally invariant. If $\|\cdot\|_2^2$ is invariant, so is $\|\cdot\|_2$, which completes the proof of equivariance for the degree-wise norm.

The squared norm of the full SPHC vector $\boldsymbol{\chi}$ transforms under rotation as

$$\begin{aligned}
\|\boldsymbol{\chi}(R_\theta\,\mathcal{R})\|_2^2 &= \left(\boldsymbol{\chi}(R_\theta\,\mathcal{R})\right)^T \boldsymbol{\chi}(R_\theta\,\mathcal{R}) \\
&= \left(\boldsymbol{D}(R_\theta)\boldsymbol{\chi}(\mathcal{R})\right)^T \left(\boldsymbol{D}(R_\theta)\boldsymbol{\chi}(\mathcal{R})\right) \\
&= \left(\boldsymbol{\chi}(\mathcal{R})\right)^T \underbrace{\left(\boldsymbol{D}(R_\theta)\right)^T \boldsymbol{D}(R_\theta)}_{=\mathbb{I}}\,\boldsymbol{\chi}(\mathcal{R}) \\
&= \|\boldsymbol{\chi}(\mathcal{R})\|_2^2\,,
\end{aligned} \tag{28}$$

where we used that the orthogonality of $\boldsymbol{D}^l$ results in orthogonality of $\boldsymbol{D}$ (as argued above).

A.2 Spherical Neighborhood

Starting point for the construction of the spherical neighborhood are the SPHCs $\mathcal{X}$ in a given layer of SO3KRATES. Consequently, the distance matrix in SPHC for all atomic pairs is given as an $n \times n$ matrix $\boldsymbol{X}$ with entries $\chi_{ij} = \|\boldsymbol{\chi}_i - \boldsymbol{\chi}_j\|_2$. The idea of a spherical neighborhood is to only consider atoms that lie within a certain distance w. r. t. each other in SPHC space, meaning only those for which $\chi_{ij} \leq \chi_{\text{cut}}$ holds. However, compared to the inter-atomic distances in Euclidean space, for which specific knowledge e.g. about bond lengths or interaction length scales exits, this is not the case for the rather abstract space of SPHCs. Thus, we apply a SOFTMAX function along each row (neighborhood) of $\boldsymbol{X}$, which gives a rescaled version of the spherical distance matrix $\tilde{\boldsymbol{X}}$ with rescaled entries $\tilde{\chi}_{ij} \in (0, 1)$. Neighborhoods for each atom are then selected based on a polynomial cutoff function [29]

$$\phi_{\chi_{\text{cut}}}(x) = 1 - \frac{(p+1)(p+2)}{2}x^p + p(p+2)x^{p+1} - \frac{p(p+1)}{2}x^{p+2}, \tag{29}$$

where the input is given as $x = \tilde{\chi}_{ij}/\chi_{\text{cut}}$. Here we chose $\chi_{\text{cut}} = \kappa/n$, where n is the number of atoms in the system and κ allows to increase or decrease the size of the cutoff radius. By scaling with the inverse of the number of atoms in the system, we ensure that the relative number of neighboring atoms remains approximately constant, even when going to larger molecules. The parameter κ allows to fine tune the relative number of atoms and can depended on the size and type of the molecule under investigation. For our cumulene experiments we chose $\kappa = 1$ and $p = 6$, which reduces the number of considered interactions by $\sim 40\%$ compared to a global model.

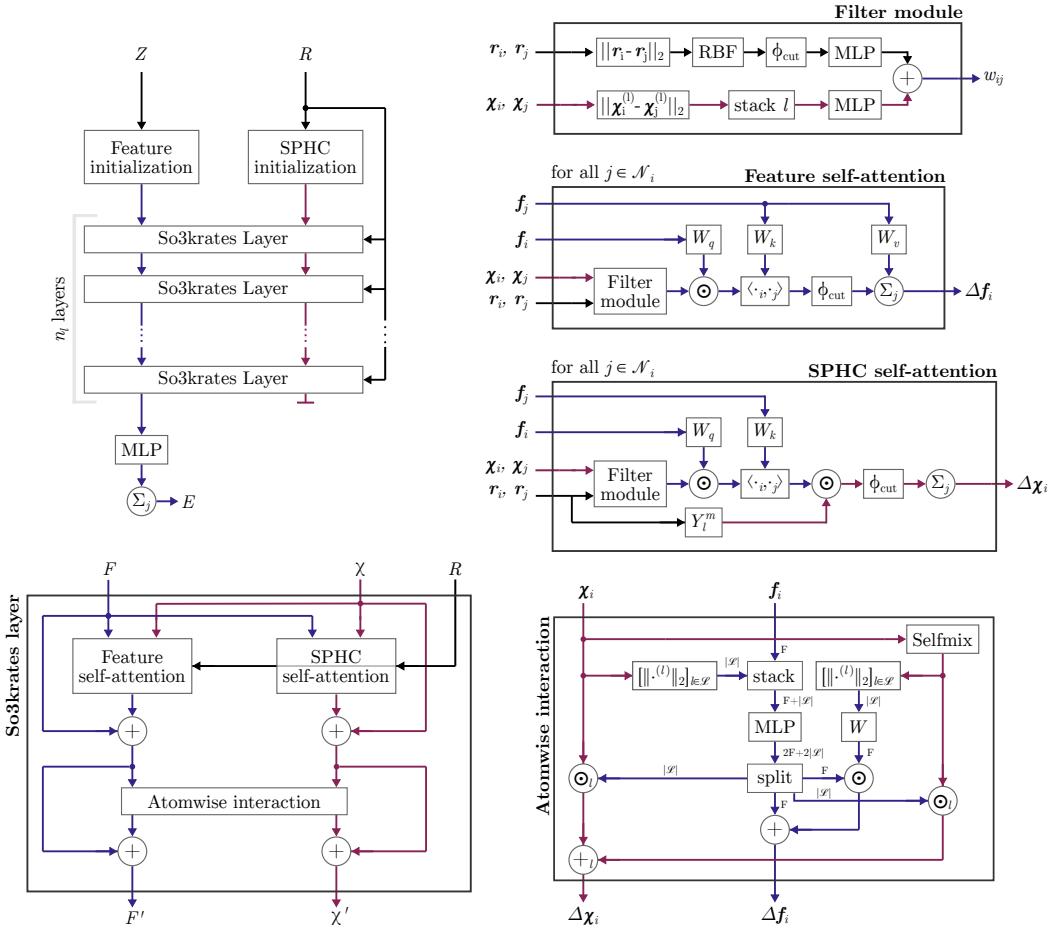

Figure 5: Figure shows the architecture of SO3KRATES and its individual building blocks. More details about the different modules can be found in the corresponding sections in A.3. Here $[\|\cdot_l\|_2]_{l\in\mathcal{L}}$ denotes per-degree norm followed by concatenation. The subscript l at the pointwise product and the addition (in the atomwise interaction module) illustrates that each degree is multiplied with one entry of the $|\mathcal{L}|$-sized output vector.

A.3 Network Architecture

The SO3KRATES model is available at `https://github.com/thorben-frank/mlff`. It makes extensive use of the libraries FLAX [55] for building the networks, OPTAX [56] for training and NUMPY [57] and JAX [58] for additional processing steps. All code and data that is necessary to produce the results step by step presented in the main body of the paper can be downloaded from `https://zenodo.org/record/6584855` (newest version).

Detailed Network Design We now describe the specific implementation of the building blocks that we have formally introduced in the main text. The computational flow of the architecture is shown in Fig. 5.

Radial Filter For the radial filter function $\phi_r : \mathbb{R} \mapsto \mathbb{R}^F$ (first summand in eq. (16)) the interatomic distances are expanded in terms of K radial basis functions as proposed in PHYSNET [19], which are given as

$$\theta^k(r_{ij}) = \phi_{r_{\text{cut}}}(r_{ij}) \cdot \exp\big(-\gamma(\exp(-r_{ij}) - \mu_k)^2\big), \tag{30}$$

where μ_k is the center of the k-th basis function ($k = 1 \dots K$). The cutoff function

$$\phi_{r_{\text{cut}}}(r_{ij}) = \frac{1}{2}\left(\cos\left(\frac{\pi r_{ij}}{r_{\text{cut}}}\right) + 1\right) \tag{31}$$

guarantees that $\theta^k(r_{ij}) = 0$ smoothly goes to zero when interatomic distances exceed the cutoff radius r_{cut}. Here we chose $K = 32$ basis functions in total. The expanded distance is passed in a two-layered MLP with SILU non-linearity in-between and 128 hidden and F output neurons, respectively.

Spherical Filter The spherical filter function $\phi_s\left([\chi_{ij}^{(l)}]_{l\in\mathcal{L}}\right) : \mathbb{R}^{|\mathcal{L}|} \mapsto \mathbb{R}^F$ (second summand in eq. (16)) is also modelled as a two-layered MLP with the same non-linearity. Its input is given as the stacked per-degree distances $\chi_{ij}^{(l)}$, such that the input to the MLP is of low dimension (number of spherical degrees l). For that reason, we only use 32 neurons in the first and again F neurons in the second layer.

Self-Attention The self-attention matrix (cf. eq. (15)) is calculated from the filter vector w_{ij} as well as from a pair of feature vectors $\boldsymbol{f}_i \in \mathbb{R}^F$ and $\boldsymbol{f}_j \in \mathbb{R}^F$. Before being passed to the inner product, each of the feature vectors is refined using a linear layer which we denote as W_q and W_k in the fashion of the key and query matrices usually appearing in the calculation of self-attention coefficients [49]. Thus, the self-attention coefficients are calculated as

$$\alpha_{ij} = \frac{1}{\sqrt{F}}\,\boldsymbol{q}_i^T(\boldsymbol{w}_{ij} \odot \boldsymbol{k}_j), \tag{32}$$

where $\boldsymbol{q}_i = W_q f_i \in \mathbb{R}^F$, $\boldsymbol{k}_j = W_k \boldsymbol{f}_j \in \mathbb{R}^F$ and $\boldsymbol{w}_{ij} \in \mathbb{R}^F$ comes from the filter module. The features $\boldsymbol{f}_j$ of the neighboring atoms are also passed through an additional linear layer W_v. The updated features are then given as

$$\boldsymbol{f}_i' = \boldsymbol{f}_i + \sum_{j\in\mathcal{N}(i)} \phi_{r_{\text{cut}}}(r_{ij}) \cdot \alpha_{ij} \cdot (W_v \boldsymbol{f}_j) \tag{33}$$

and the updates to the SPHCs as

$$\boldsymbol{\chi}_i'^{(l)} = \boldsymbol{\chi}_i^{(l)} + \sum_{j\in\mathcal{N}(i)} \phi_{r_{\text{cut}}}(r_{ij}) \cdot \alpha_{ij}^{(l)} \cdot \boldsymbol{Y}^{(l)}(\boldsymbol{r}_{ij})\,. \tag{34}$$

Different parameters are used for the feature and SPHC updates, respectively. For the feature update, we use a predefined number of heads whereas the number of heads in the SPHC update equals the number of degrees in the SPHC vector. In order to ensure permutation invariance, all parameters of the linear layers are shared across atoms.

Atomwise Interaction After the update MP step, we update the features as well as the SPHCs per atom. In this step, we do not only include cross-degree coupling in $\boldsymbol{\chi}$ but also allow for information exchange between the feature and the SPHC branch. The functions $\phi_1\left(\mathbf{f}_i, [\chi_i^{(l)}]_{l\in\mathcal{L}}, [\tilde{\chi}_i^{(l)}]_{l\in\mathcal{L}}\right)$ and $\phi_2^{(l)}\left(\mathbf{f}_i, [\chi_i^{(l)}]_{l\in\mathcal{L}}, [\tilde{\chi}_i^{(l)}]_{l\in\mathcal{L}}\right)$ are implemented by a shared MLP. The function $\phi_3^{(l)}\left([\tilde{\chi}_i^{(l)}]_{l\in\mathcal{L}}\right)$ is implemented by a single linear layer, without bias term. The full computational flow is shown in Fig. 5.

A.4 Ablation Study Spherical Filter

To illustrate the importance of the spherical filter, we examine its effect in an ablation study on the MD17 benchmark for varying maximal degree $l_{\max}$. As can be seen in Fig. 6, using spherical filters (see eq. (16)) improves performance compared to a SO3KRATES model without them (solid vs. dotted lines), where the difference becomes even more pronounced for force predictions. Further, the effect of spherical filters becomes stronger for smaller $l_{\max}$. Since many equivariant MPNNs carry features including additional channels for higher order geometric information, spherical filters can be straight forwardly integrated into current architectures, offering the potential of increased accuracy.

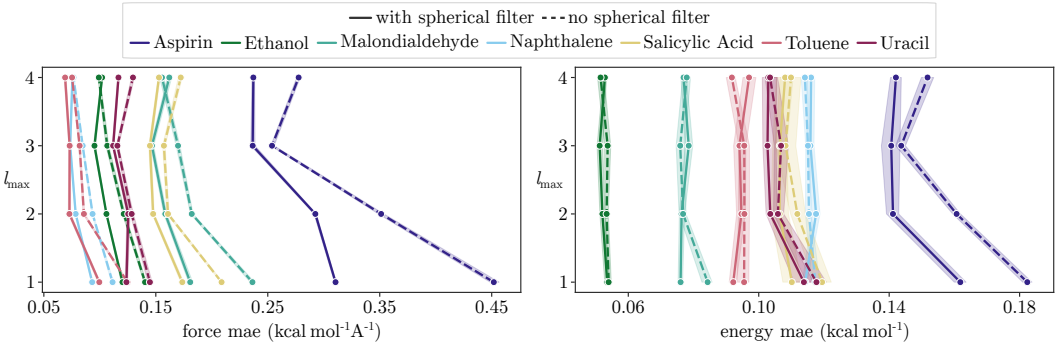

Figure 6: Ablation study on MD17 for using spherical filters. The model is evaluated on five different splits (a 1k points) of the data, where the transparent envelope is the 2σ confidence interval.

Table 2: Energy and force MAEs in meV and meV Å^{-1} for different models trained on the full QM7-X data set.

		SchNet [13]	PaiNN [31]	SpookyNet [30]	So3krates $F = 132$	So3krates $F = 264$	So3krates + *non local*
known molecules / unknown conformations	*Energy*	50.847	15.691	10.620	16.815	15.228	16.176
	Forces	53.695	20.301	14.851	20.422	18.446	19.879
unknown molecules / unknown conformations	*Energy*	51.275	17.594	13.151	21.733	21.750	20.071
	Forces	62.770	24.161	17.326	25.211	23.169	24.236

A.5 QM7-X Experiments

As an additional benchmark, we train So3krates on the full QM7-X data set and compare our results to the reported errors for known and unknown molecules in table 2. The known molecules correspond to the test samples that have not been seen during training. Following [30], we use the keys (idmol in the QM7-X data base) [1771, 1805, 1824, 2020, 2085, 2117, 3019, 3108, 3190, 3217, 3257, 3329, 3531, 4010, 4181, 4319, 4713, 5174, 5370, 5580, 5891, 6315, 6583, 6809, 7020] for testing the generalization of our model to unknown structures, which have been excluded from the data set prior to generating train, test and validation splits.

Accuracy Comparison Local vs. Non-Local Model By comparing the results reported in table 2 we find, that the non-local model shows the same accuracy as the local model for both known and unknown molecules. This underlines the applicability of the presented, non-local corrections to a large variety of molecular structures including transfer learning to unknown molecules.

Accuracy Improvement by Up-Scaling Despite the parametric leightweight structure of So3krates, it is important that results can be imrproved by upscaling the model. Here we take one of the most straight forward paths and upscale the model by simply increasing the feature dimension from $F = 132$ to $F = 164$. As it can be seen in table 2, this already allows to increase the accuracy on the QM7-X dataset compared to the base line model. It should be noted, however, that one could follow additional/other directions, such as increasing the maximal degree $l_{\max}$.

Generalization to Larger Molecules with a Non-Local Model As seen in table 2, a model with non-local correction is capable of generalizing to structures not seen during training. However, the re-usability of local information is one of the key assumptions for building models that can be trained on small molecules and afterwards be applied to larger, completely unknown structures [30]. Thus, testing the non-local model on unknwon structures from the QM7-X data set is an insufficient test to investigate this property of transfer learning. We therefore use the non-local model for geometry optimization of molecules that range from 47 (riboflavin) up to 109 atoms (Ala10) in size (see 7). We find that the non-local model is robust for geometry optimization when being applied to much larger, unknown molecules. The largest molecule in the QM7-X data set has 23 atoms.

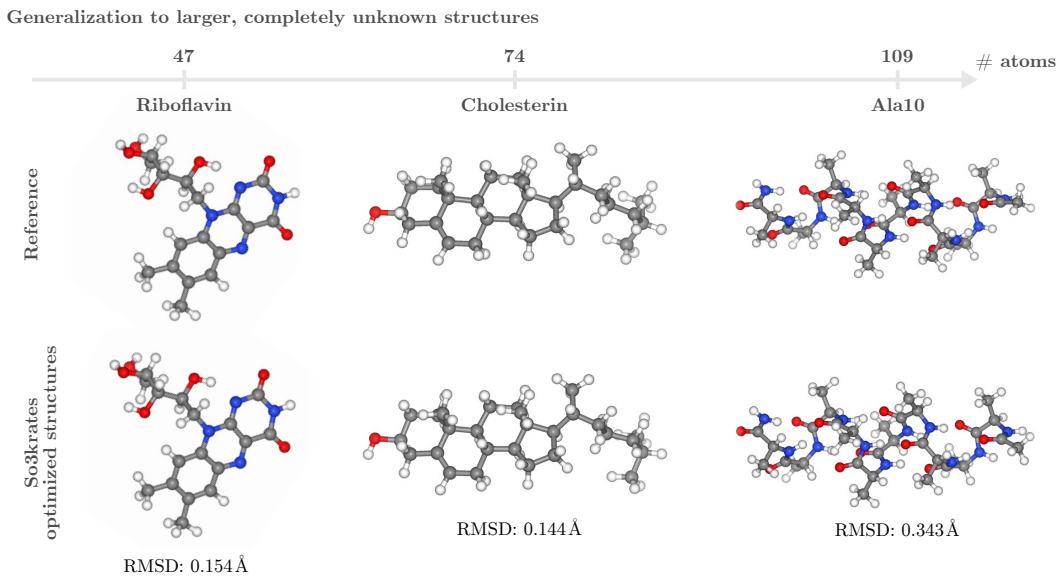

Figure 7: The upper row shows the optimized reference geometry obtained with PBE0-MBD calculation (taken from [30]). The bottom row shows geometry optimized structures obtained with a SO3KRATES model *with* non-local corrections from SPHC space as well as the root mean squared deviation from the reference structure. The model has been trained on the QM7-X data set for which the largest contained structure has 23 atoms. Thus, SO3KRATES can generalize to larger, unknown structures without relying on the assumption of locality.

Table 3: Training and inference times for different models for the toluene molecule from the MD17 benchmark and a batch size of 4. Hyperparemters for SO3KRATES are the ones that have been used to produce the reported results on the MD17 benchmark (cf. Tab. 1). Times for DIMENET and GEMNETQ have been measured on a different GPU, such that we decrease their runtimes by the factor reported here [37]. Inference times are for energy and force predictions.

	SO3KRATES		NEQUIP [32]		DIMENET [29]		GEMNETQ [34]	
	training	*inference*	*training*	*inference*	*training*	*inference*	*training*	*inference*
time (ms)	34	12	507	136	218	24	483	76

A.6 Time Analysis

In order to determine the training and inference times, we follow [34] and evaluate our model on the toluene molecule from the MD17 benchmark with a batch size of 4. We trained a NEQUIP model with the same hyperparameters as reported in [32] as well as a SO3KRATES model on a Tesla P100 with 12GB. The reported training time corresponds to the wall time it took each model to evaluate a single gradient update (without time for validation). We compare these runtimes to the runtimes that have been reported for DIMENET [29] and GEMNETQ [34] in [34]. However, these reported times have been measured on a GeForce GTX 1080Ti (a GPU we did not have access to). In [37] it has been found that a Tesla P100 gives a speedup factor of ~ 1.3, such that we downscale the reported runtimes accordingly. The resulting times are shown in table 3, which are the values plotted in 1d. It should be noted, that our implementation did not focus on the runtime, such that it is likely to be possible to further reduce the computational cost that is required for training and inference.

A.7 Analysis of Attention Coefficients

In figure 8 we plot the attention coefficients from the MP update of the SPHCs (cf. eq. (14)) after training for different dihedral angles between the rotors. Attention values are calculated as the average over all attention values obtained for a given atomic pair throughout all SPHC update steps. As it can be verified visually, the model picks up physically important interactions. Care should be taken,

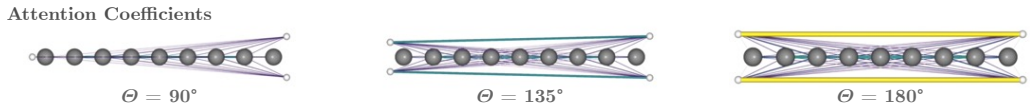

Figure 8: Visualization of the attention coefficients from the MP update of the SPHCs (cf. eq. (14)) for different dihedral angles between the rotors.

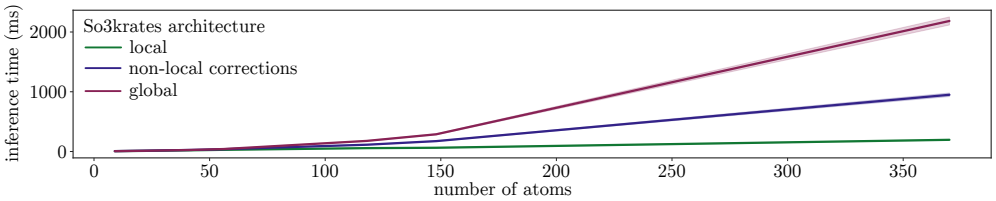

Figure 9: Scaling comparison between a local SO3KRATES (as used in the first part of the experiment section 4.1), an architecture with non-local corrections (as used in the second part of the experiment section 4.2) and a global SO3KRATES model (only used for the sake of comparison). As expected, the local model shows linear scaling in the number of atoms and only takes ~ 196 ms for a system with 370 atoms. We also find the model with non-local correction to balance between the quadratically scaling global model and the linear local model. Due to memory issues for the fully global model for larger structures, we only report inference times for energy predictions.

however, since it has been pointed out in [59] that looking at the bare attention values only has limited expressiveness.

A.8 Scaling Analysis

We further compare the scaling of three different versions of the SO3KRATES model. Namely, we compare a local SO3KRATES model, which only localizes in $\mathbb{R}^3$ (used in 4.1), a non-local SO3KRATES model with non-local corrections from SPHC space (used in 4.2) and a fully global model for which the cutoff radius is chosen such that all atoms are in each others neighborhood. The inference times for energy predictions for molecules ranging from 9 up to 370 atoms are shown in Fig. 9. As expected, we find linear scaling for the local SO3KRATES version with a remarkable inference time of only ~ 196 ms for 370 atoms (batch size is 25). For the fully global mode, we see quadratic scaling in the number of atoms. The model with non-local corrections can be found somewhere in-between the local and the global model. To that end, we want to stretch the fact that we did not focus on an efficient implementation for the non-local corrections. The experiments were run on a Tesla P100 with 12 Gb and a batch size of 25.

A.9 QM7-X250

As starting point for the recently introduced QM7-X dataset [51] serve $\sim$ 7k molecular graphs with a maximum of 7 non-hydrogen atoms (C, N, O, S, Cl). By sampling and optimizing structural and constitutional isomers for each graph, ~ 42k equilibrium structures are generated. Using normal mode sampling at 1500 K, 100 out-of-equilibrium points are generated for each structure resulting in 101 data points per structure and ~ 4.2M geometries in total.

In order to make the data set well suited for both, kernel and neural network models, we group the geometries by structural isomers which gives $\sim$ 13k individual data sets, each consisting of *#stereo-isomers* $\times 101$ geometries. For each of the data sets we choose 80 points for training, 10 points for validation and the remaining points for testing after the training. Afterwards, we randomly sample 250 data sets out of the 13k data sets. The comparison of the probabilities of drawing a molecule with a given number of atoms, number of symmetries and number of stereo-isomers from the original QM7-X and the QM7-X250 data sets are shown in Fig. 10. Since we ensure that each of the structural subsets present in the original data set is also present in the 250 drawn samples at least once, it can be seen that these structures are over represented in the QM7-X250 data set, even though

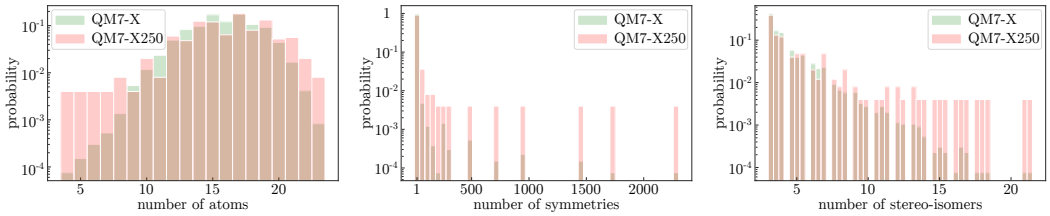

Figure 10: The figure shows the probability of occurrence for structures with a certain property value in the original QM7-X and the sub-sampled QM7-X250 dataset. As it can be seen, QM7-X follows the distribution of the original dataset. As for each property value at least once structure is present in QM7-X250 (per design), values with a low count in the original data set have higher relative importance, which can be seen e.g. for structures with only a few atoms, or with high symmetry.

Table 4: Comparison of the averaged per structure MAE for energy (in meV) and forces (in meV Å^{-1}) for sGDML and SO3KRATES models with varying $l_{\max}$. Best results are in bold. Note that for small $l_{\max}$ the total number of parameters decreases, since in the SPHC update step, as many heads are used as there are degrees (one degree for $l_{\max} = 0$ and $l_{\max} = 1$). As a consequence, the size of the matrices Q and K, that are applied to the feature vectors that make up the inner product, become smaller in size. The feature dimension $F = 132$ is the same for all models. This leads to a decrease of the total number of network parameters, which is larger than the increase in network parameters, due to the additional degrees. Note that it is important to take the average over per structure MAEs, since the different structures have differently many test samples (see section A.9).

		sGDML [38]		SO3KRATES $l_{\max} = 0$		SO3KRATES $l_{\max} = 1$		SO3KRATES $l_{\max} = 2$		SO3KRATES $l_{\max} = 3$	
		individual	*joint*	*individual*	*joint*	*individual*	*joint*	*individual*	*joint*	*individual*	*joint*
QM7-X250	*energy*	67.78	–	78.40	46.87	66.43	44.64	47.02	19.10	**38.40**	**17.09**
	forces	107.66	–	105.80	57.57	84.54	52.58	59.33	27.77	**48.46**	**25.37**
# parameters		–		846k		846k		746k		716k	

they only make up a single structure. Apart from these special structures, we see that the sampled dataset correctly reproduces the distribution of the original data set.

From the 250 sampled structures, we build 250 per structure data sets with 80 training, 10 validation and 10–3748 (depending on the number of stereo-isomers) test points, which we referred to as *individual* dataset in the main text. We can further build a *joint* dataset, by merging all individual structures into a single data set, which gives 20 000 (250×80) training, 2 500 (250×10) validation, and 108 800 testing points.

A.10 QM7-X250 Experiments

In the main text we show the force MAE as a function of maximal order $l_{\max}$ in Fig. 4a. In table 4, we further show the exact errors for energy and forces as a function of maximal order $l_{\max}$, as well as the number of parameters per SO3KRATES model. For all models with $l_{\max} \geq 1$, we do not include $l_{\max} = 0$ within the SPHCs, since the zeroth degree evaluates to a constant one and thus does not contain any additional geometric information. The generalization to unknown molecules is tested on the same structure keys as in the SPOOKYNET paper [30] and in the QM7-X experiment from above A.5. As we only trained on forces and generalize to completely unknown molecules, we can not fit the energy integration constant (as we assume to have no reference data for the unknown molecules). For that reason we only report force errors (see 5). However, one could train a SO3KRATES model on both, energy and forces to obtain meaningful predictions for both. In that case, atomization energies would need to be included to obtain equal energy scales across different molecules as done for the full QM7-X data set.

Table 5: Comparison of the averaged per structure MAE for forces (in meV Å^{-1}) for sGDML and SO3KRATES models with varying $l_{\max}$ when applied to completely unknown structures. Best results are in bold. Due to the way the molecular descriptor is designed, sGDML [38] models can only be trained on individual structures. Thus the reported results for sGDML are not generalization results but rather serve as a benchmark for the generalization MAE of SO3KRATES. The number of parameters are the same as reported in Tab. 4. Note that it is important to take the average over per structure MAEs, since the different structures have differently many test samples (see section A.9).

		sGDML [38]	SO3KRATES $l_{\max}=0$	SO3KRATES $l_{\max}=1$	SO3KRATES $l_{\max}=2$	SO3KRATES $l_{\max}=3$
Generalization	*forces*	86.84	159.42	117.91	76.05	**68.29**

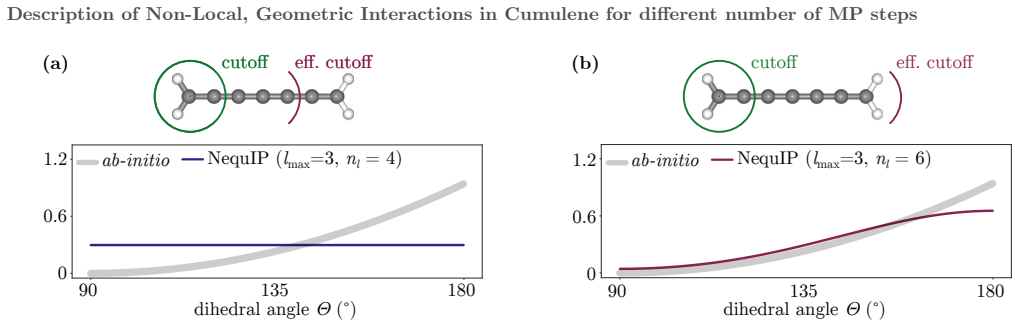

Figure 11: **(a)** Description of the energy profile in cumulene using $n_l = 4$ in the NEQUIP model, such that the effective cutoff radius is smaller then the length scale on which the electronic effects take place. **(b)** By increasing the number of layers to $n_l = 6$, the effective cutoff becomes large enough to transfer information about the rotor orientation along the molecular graph. However, the problem is only shifted to larger length scales.

A.11 Additional Experiments for Non-Local Effects

Here we demonstrate, that increasing the number of local MP steps in NEQUIP allows to model the non-local effects in cumulene, as an increase of the number of steps increases the effective cutoff of the model (see Fig. 11). However, this just shifts the problem to larger distances; increasing the length of the cumulene molecule again leads to a scenario where the local model fails.

A.12 Benchmarks for Non-Local Effects

We further apply SO3KRATES to the benchmark presented in [60], which explicitly introduces structures which exhibit non-local effects. We find, that for the carbon chain with non-local charge transfer, SO3KRATES with non-local corrections improves by a factor ~ 2 compared to a local model. For the remaining structures no such difference can be found. Notably, the non-local effects from the presented benchmark do not result in global geometric relations but rather change the overall behavior by adding or removing certain atoms (called doping). Thus, a combination of the SO3KRATES mechanism (for geometric relations) and the mechanism presented in SPOOKYNET might be a valuable direction for future work. Full results are reported in table 6. Note that SO3KRATES could be easily extended to predict partial charges as well following e.g. [30].

A.13 Training Details

We train SO3KRATES by minimizing a combined loss of energy and forces

$$\text{Loss} = (1-\beta)\cdot(E-\tilde{E})^2 + \frac{\beta}{3N}\sum_{k=1}^{n}\sum_{i\in(x,y,z)}(F_k^i-\tilde{F}_k^i)^2, \tag{35}$$

Table 6: RSME errors for energy and forces in meV / atom and meV / Å for various generations of Behler-Parinello networks [60], SPOOKYNET [30] and a local and non-local SO3KRATES model.

		2G-BPNN	3G-BPNN	4G-BPNN	SPOOKYNET	SO3KRATES	SO3KRATES + *non local*
$C_{10}H_2$ / $C_{10}H_2^+$	*Energy*	1.619	2.045	1.194	0.364	0.113	0.122
	Forces	129.5	231.0	78.0	5.802	13.198	7.844
	Charges	–	20.08	6.577	0.117	–	–
$Na_{8/9}Cl_8^+$	*Energy*	1.692	2.042	32.78	1.052	0.455	0.474
	Forces	57.39	76.67	15.82	1.052	3.126	3.316
	Charges	–	20.80	1.323	0.111	–	–
Au_2-MgO	*Energy*	2.287	–	0.219	0.107	0.062	0.064
	Forces	153.1	–	66.0	5.337	8.130	9.472
	Charges	–	–	5.698	1.013	–	–

Table 7: **Training hyperparameters:** Table summarizes the training hyperparameters, used for the different models and experiments. Experiments are identified via their figure/table number in the main text. For the results reported in Tab. 6, the same sizes of data splits as reported in [60] have been used (indicated by an asterix).

Ref.	F	n_{layers}	r_{cut} (Å)	l_{max}	geom. corr.	spherical filter	β	epochs	B_s	N_{train}	N_{valid}
Fig. 2	128	4	2.5	1	True/False	True	1	4k	8	1k	1k
Fig. 3	128	4	2.5	1	True/False	True	1	4k	8	1k	1k
Fig. 4a	132	6	5	[0,1,2,3]	False	True	1	1.5k	1	80	10
Fig. 4b	132	6	5	[0,1,2,3]	False	True	1	6k	100	20k	2.5k
Tab. 1	132	6	5	3	False	True	0.99	4k	8	1k	1k
Fig. 6	[128, 128, 132, 128]	6	5	[1,2,3,4]	False	True / False	0.99	4k	8	1k	1k
Fig. 7	132	6	5	3	True	True	0.99	1k	100	3.6M	360k
Tab. 2	132	6	5	3	True/False	True	0.99	1k	100	3.6M	360k
Tab. 2	264	6	5	3	False	True	0.99	1k	100	3.6M	360k
Tab. 6	132	6	5	3	True/False	True	0.99	1k	100	*	*

where $\tilde{E}$ and $\tilde{F}$ are the ground truth and E and F are the predictions of the model. The loss is evaluated on mini batches with a batch size given in table Tab. 7. The parameter β is used to control the trade-off between energy and forces and additionally accounts for different energy and force scales. We train our models with the ADAM optimizer [61] and an initial learning rate of $\mu = 1 \times 10^{-3}$. We use exponential learning rate decay where the learning rate is decreased by a factor of 0.5 every 1k epochs for the MD17 benchmark and the joint QM7-X250 dataset and every 300 epochs for individual models on the QM7-X250 dataset. For training on the full QM7-X data set we reduced the leraning rate every 250 epochs by a factor of 0.7. We further applied gradient clipping to a maximal norm of 1.

Additional hyperparameters that have been used to produce the tables and figures in this work are given Tab. 7. Whenever $\beta = 1$ is reported, no energy contribution did enter the loss function. In that case, we calculated the integration constant for energy according to section A.14.

A.14 Energy Integration Constant

When only training on forces, the resulting energy predictions are likely to be shifted w. r. t. the correct energy values, due to vanishing constants when taking the derivative. Since force fields are conservative vector fields, one can define the following loss for the constant c as

$$\begin{aligned} \mathcal{L}(c) &= \sum_{i=1}^{M} \left(\int \mathbf{f}_{\mathrm{F}}(R_i)\, \mathrm{d}R_i - E_i \right)^2 \\ &= \sum_{i=1}^{N_{\text{data}}} \left(f_{\mathrm{E}}(R_i) + c - E_i \right)^2 \end{aligned} \tag{36}$$

where index i runs over the M data points, $R_i \in \mathbb{R}^{n\times 3}$ are the atomic coordinates and E_i is the reference value of the PES. The functions $\mathbf{f}_\mathrm{F} : \mathbb{R}^{n\times 3} \mapsto \mathbb{R}^{n\times 3}$ and $f_\mathrm{E} : \mathbb{R}^{n\times 3} \mapsto \mathbb{R}$ are the force and energy function, respectively. Minimization w. r. t. to c then gives

$$\partial_c \mathcal{L}(c) \stackrel{!}{=} 0 \tag{37}$$

$$\iff \qquad 2\sum_{i=1}^{M} c - \big(E_i + f_\mathrm{E}(R_i)\big) = 0 \tag{38}$$

$$\iff \qquad \frac{1}{M}\sum_{i=1}^{M} E_i + f_\mathrm{E}(R_i) = c\,. \tag{39}$$

Thus, the shifted energy function is given as $\tilde{f}_\mathrm{E}(R_i) = c + f_\mathrm{E}(R_i)$.