# A Reactive, Scalable and Transferable Model for Molecular Energies from a Neural Network Approach Based on Local Information

Oliver T. Unke* and Markus Meuwly*

*Department of Chemistry, University of Basel, Klingelbergstrasse 80, CH-4056 Basel, Switzerland.*

E-mail: oliver.unke@unibas.ch; m.meuwly@unibas.ch

### Abstract

Despite the ever-increasing computer power, accurate *ab initio* calculations for large systems (thousands to millions of atoms) remain infeasible. Instead, approximate empirical energy functions are used. Most current approaches are either transferable between different chemical systems, but not particularly accurate, or they are fine-tuned to a specific application. In this work, a data-driven method to construct a potential energy surface based on neural networks is presented. Since the total energy is decomposed into local atomic contributions, the evaluation is easily parallelizable and scales linearly with system size. With prediction errors below 0.5 kcal mol$^{-1}$ for both, unknown molecules and configurations, the method is accurate across chemical and configurational space, which is demonstrated by applying it to data sets from nonreactive and reactive molecular dynamics simulations and a diverse database of equilibrium structures. The possibility to use small molecules as reference data to predict larger structures is also explored. Since the descriptor only uses local information, high-level *ab initio* methods, which are computationally too expensive for large

molecules, become feasible for generating the necessary reference data used to train the neural network.

# 1 Introduction

In 1929 Paul Dirac[1] noted that the (electronic and nuclear) Schrödinger equation (SE) contains all that is necessary to describe chemical phenomena and processes. As the underlying equation (SE) is too complicated to be solved in closed form but for the simplest systems, computational and numerical methods have been devised to find approximate solutions such that meaningful information about a system and/or a process can be obtained. Depending on the observable of interest, the meaning of "accuracy" may change, though. A total number of several ten thousand atoms is "large" from the perspective of what system size can be realistically investigated at the single-point energy level using density functional theory (DFT).[2] With increasing accuracy, or when considering optimized structures, vibrations or even (classical) nuclear dynamics, the size of the system that is computationally tractable by explicitly solving the electronic SE (i.e. by "ab initio" rather than semiempirical methods) reduces to less than thousand atoms.[3] These limitations have spurred the development of alternative, more empirical methods.

For small systems (few atoms) it is common practice to directly interpolate a set of known and precomputed reference energies (obtained from a pointwise solution of the electronic SE) to construct a continuous functional form. Popular interpolation techniques include the modified Shepard algorithm,[4–6] the moving least-squares method,[7–9] permutational invariant polynomials[10–12] and the reproducing kernel Hilbert space method.[13–16]

For big systems (proteins or condensed matter) a typical approach is to fit a large number ($> 10^3$) of parameters of an empirical functional form, a so-called force field (FF), either to best reproduce reference energies, experimental data that can be derived from them (e.g. thermodynamic or spectroscopic observables) or both.[17] While some parameters can be determined from experiment, others (e.g. partial atomic charges) require electronic structure calculations for fragments or explicit molecular dynamics (MD) simulations (e.g. van der

Waals parameters). Once parametrized, the total energy and corresponding forces required for MD simulations can be evaluated much more efficiently than by directly (and approximately) solving the SE.[18,19] With currently available computer power it is, for example, possible to run explicit atomistic MD simulations for small parts of a cell for several 100 ns.[20] However, general empirical FFs[21–25] also have a number of drawbacks,[26] including their limited accuracy, or the fact that most of them do not allow bond-breaking/bond-formation to be described. Although it is now possible to parametrize a FF to within fractions of 1 kcal mol$^{-1}$ (for energies) for single, isolated systems and special potentials for metals,[27–32] bond-order based (reactive) potentials,[33–36] and reactive force fields for particular systems[37–41] or processes (e.g. proton transfer),[42] have become available, it would be desirable to generalize this to larger classes of problems, irrespective of the particular type of application one has in mind.

One possible step in this direction has been taken during the past decade when machine learning (ML) approaches, which give computers the ability to learn without being explicitly programmed,[43] have been applied to train a computer system using large amounts of precomputed data (typically energies) to estimate properties for unknown compounds or structures.[44–47] Hence, instead of approximately solving the electronic SE or representing its solution through a ball-and-spring model as in a FF, a computer system learns to predict energies based on an increasing amount of data. Such an approach is motivated by the observation that the electronic Hamiltonian $\hat{H}$ is uniquely determined by the external potential, which in turn depends only on the set of nuclear charges $\{Z_i\}$ and atomic positions $\{\mathbf{r}_i\}$ of the system. Therefore, all information necessary to determine $E$ is contained in $\{Z_i, \mathbf{r}_i\}$ and there must exist an exact mapping $f : \{Z_i, \mathbf{r}_i\} \mapsto E$, which returns the energy $E$ given $\{Z_i, \mathbf{r}_i\}$. If the mapping $f(Z_i, \mathbf{r}_i)$ was known, directly *solving* the SE could be *circumvented*. This situation is reminiscent of density functional theory (DFT) in that the existence of a suitable functional is guaranteed but its actual form is not known. As such, the fundamental object of interest in the present work is the potential energy surface (PES), an approximation to $f : \{Z_i, \mathbf{r}_i\} \mapsto E$, which corresponds to a $3N-$dimensional hypersurface that returns the total potential energy of a system $E_{\mathrm{tot}}(\mathbf{r}_i)$ given the positions $\mathbf{r}_i$ of all $N$ nuclei.

Artificial neural networks (NNs)[48–54] are a popular class of ML algorithms which have been used to tackle various difficult problems, including speech,[55] image[56] and face recognition.[57] In particular, feed-forward NNs have been proven to be general function approximators,[58,59] which makes them suitable for approximating $f : \{Z_i, \mathbf{r}_i\} \mapsto E$. Ideally, the resulting PES should be accurate, rapid to evaluate, analytically differentiable, systematically improvable, scalable and applicable to bond-breaking/bond-formation problems ("reactive PES"). Additionally, it should be transferable between different systems and configurations.[60] Existing PESs typically fulfill only some of these requirements and the "ideal PES" does not exist yet, probably due to the difficulty of finding a functional form that would satisfy all needs simultaneously. In contrast, NNs do not assume a predefined functional form, and could offer important advantages.

NNs have been used previously to fit PESs for molecular systems in the spirit of many-body expansions.[61–64] While being accurate, they typically involve a large number of individual NNs (one for each term in the many-body expansion), making the method scale poorly for large systems. Recently, there have also been efforts to predict bond energies using a NN.[65]

An alternative approach, known as high-dimensional NN (HDNN) and first proposed for bulk silicon,[66] decomposes the total energy of a system into atomic contributions, which is appealing, because "energy" is an extensive property and it allows to apply the same network to systems of different size. In an HDNN, an atomic descriptor vector (the "fingerprint for the atomic environment") is provided as input and yields the atomic energy $E_i$ as output. All atomic contributions are added to give the total energy $E_{\mathrm{tot}}$ of the system for a particular configuration $\{\mathbf{r}_i\}$.

It is useful to introduce an atomic descriptor because the dimensionality of the input vector $\mathbf{x}_{\text{in}}$ needs to be fixed in a feed-forward NN and using Cartesian coordinates as input would limit the applicability of the network to specific system sizes. The descriptor combines the influence of all neighboring atoms up to a cutoff radius $R$ (e.g. 6 Å)[60] with a continuous behaviour at the boundary. Introducing a cutoff allows the method to scale linearly with respect to the number of atoms. Another disadvantage of using Cartesian coordinates is that they are not invariant with respect to translation and rotation. Since NNs are purely numerical algorithms, they would output different values if the input coordinates changed due to such transformations of the system. In contrast, the descriptor is designed to be identical for all symmetry equivalent representations by construction.

In an HDNN, the entries of the descriptor vector are the values of several so-called symmetry functions, which algebraically combine distances and/or angles between the atom of interest and all other atoms in its neighborhood such that the resulting value is invariant with respect to translation, rotation and permutation of equivalent atoms. The individual symmetry functions are manually designed to respond differently to distinct combinations of distances and/or angles, such that a sufficient number ($\approx$ 50)[66] of symmetry functions provides a unique fingerprint for an atomic environment.[60]

Alternative methods to construct atomic environment descriptors as input for a NN based on orthonormal 3-D Zernike basis functions[67] or radial and angular distribution functions[68] have been discussed in the literature. In contrast, the smooth overlap of atomic positions (SOAP) approach[69] directly introduces a distance metric and a similarity kernel for atomic environments, such that it is not necessary to explicitly calculate the descriptor. Therefore, the SOAP approach is more suited for kernel-based ML methods.[70]

In order to apply HDNNs to multi-component systems,[71] the symmetry functions are duplicated for each species and a separate NN is trained for each element.[60] Unfortunately, due to the rapidly increasing complexity of chemical space, this approach is still limited to systems containing only few chemical elements.[72] Furthermore, such NNs are not transferable across chemical space and have to be retrained for every new system of interest.

A conceptually different approach, the deep tensor NN (DTNN),[73] allows to reuse the same NN to predict energies of systems with different composition across chemical space. Similar to HDNNs, the DTNN accumulates atomic energy contributions to predict the total energy $E_{\text{tot}}$. However, instead of an environment descriptor based on symmetry functions, it receives a vector of nuclear charges and a matrix of atomic distances as input. A tensor layer[74–76] then builds a coefficient vector $\mathbf{c}_i$ for each atom $i$, which acts as the environment-dependent fingerprint. To do so, the coefficient vector $\mathbf{c}_i$ is initialized depending on the species of atom $i$ and recursively refined in $T$ steps by adding interaction vectors $\mathbf{v}_{ij}$, which depend on the pairwise distance between atoms $i$ and $j \neq i$, as well as the current $\mathbf{c}_j$ of atom $j$. After $T$ refinements, the final $\mathbf{c}_i$ is passed as input to a fully-connected layer to determine the atomic energy contribution $E_i$ of atom $i$.

Because each refinement step considers all pairwise distances, the evaluation of the DTNN scales quadratically with respect to the number of atoms. Although introducing a distance cutoff to achieve linear scaling has been proposed,[73] it is important to note that even with a cutoff, the network still requires information about all atoms present in the system in order to recursively refine the coefficient vectors $\mathbf{c}_i$ (every refinement step requires knowledge about the current $\mathbf{c}_j$ of other atoms). Using $T = 3$ interaction passes and 100k reference structures, the DTNN predicts the energy of structures in the QM9 dataset[77] accurately with a mean absolute error (MAE) of 0.84 kcal mol$^{-1}$.[73]

More recently, the SchNet architecture was proposed,[78] which improves upon the DTNN. Instead of refining the coefficient vectors $\mathbf{c}_i$ with a tensor layer, they are iteratively updated by residual connections[79] between three interaction blocks.[78] The interaction blocks utilize interatomic continuous-filter convolutions[78] and fully-connected layers to couple different coefficient vectors based on pairwise interatomic distances. The final coefficient vectors $\mathbf{c}_i$ are passed through two fully-connected layers, which output atomic energy contributions $E_i$. Similar to the DTNN, SchNet requires information of all atoms present in the system to update the coefficient vectors $\mathbf{c}_i$, even if a cutoff radius was introduced (the current version of SchNet does not employ a cutoff).[78] When trained on 100k reference structures, SchNet predicts the energy of structures in the QM9 dataset[77] with a MAE of 0.34 kcal mol$^{-1}$. For a more detailed description of SchNet, the reader is referred to ref. 78.[80]

Because both, DTNN and SchNet, require global information about all atoms in a system for the iterative refinement of $\mathbf{c}_i$, individual atomic contributions cannot be evaluated independently without communicating intermediate results. While such approaches are the method of choice for individual molecules or small systems, it might be difficult to apply them routinely to condensed phase systems containing thousands of atoms with a multitude of chemical environments such as in proteins.

In the present work, a NN-based method tailored for accurate energy evaluations, which can be applied to construct PESs for nonreactive and reactive dynamics of chemically heterogeneous systems in the condensed phase, is introduced. While being inspired by high-dimensional NNs, the descriptor does not rely on hand-crafted symmetry functions and encodes atomic species and environment simultaneously, similar to the coefficient vectors $\mathbf{c}_i$ in the DTNN and SchNet. This allows to train a single NN to predict the atomic energy contributions $E_i$ of all elements in their chemical environments. In contrast, high-dimensional

NNs require separate NNs for each element. Contrary to iterative approaches based on tensor layers[73] or convolution,[78] the descriptor contains strictly local information and is calculated in a single step. Thus, the proposed method scales linearly with respect to system size and can even be evaluated in parallel, because each atomic descriptor is independent of other descriptors and needs no iterative refinement. When applied to the QM9 dataset,[77] the proposed approach yields predictions with errors below 0.5 kcal mol$^{-1}$, transferable across chemical space. The predictions are also transferable across configurational space, as is demonstrated by applying the same method to several MD datasets.[81] When trained with appropriate reference data, the method is also able to describe reactions. By analyzing individual atomic energy contributions $E_i$, it is shown that the network predicts energies in a chemically intuitive and interpretable way. Further, the possibility to train the network on small molecules to predict the energies of larger systems is demonstrated. Finally, possible future improvements are discussed.

# 2 Methods

In order to predict the energy of a system of interest, such as a molecule, a descriptor for each atom is supplied to a NN, which predicts an atomic energy contribution $E_i$. The individual contributions are added to obtain the total energy $E_{\text{tot}}$. Figure 1 gives a schematic overview of the computational protocol.

In the following, the atomic descriptor (section 2.1), the NN (section 2.2) and the process for training the NN (section 2.3) are described in more detail. It is important to note that only total energies are required as reference data during training, as the NN automatically learns to perform the energy decomposition into atomic contributions. This way, only true quantum mechanical observables are used as reference data and no, ultimately arbitrary,

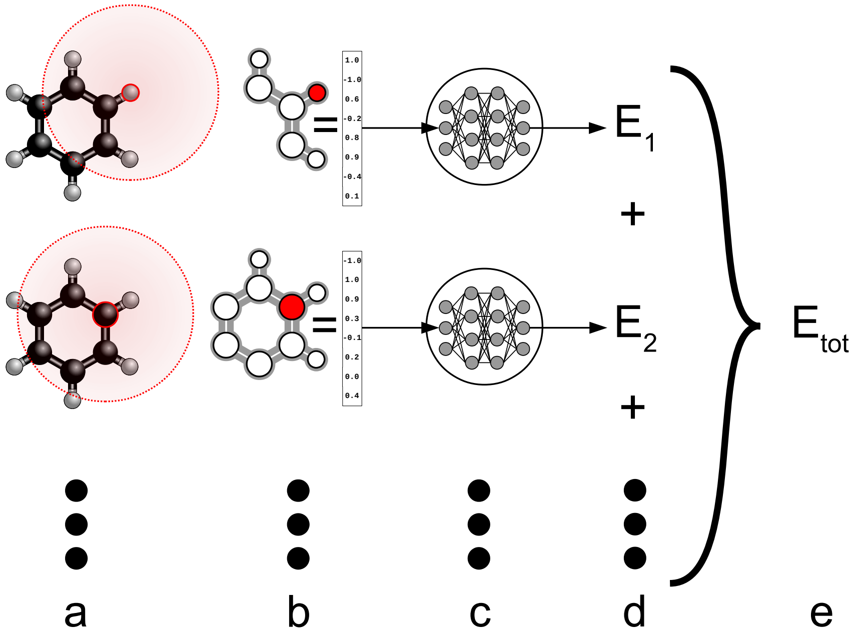


Figure 1: Schematic representation of predicting the energy $E_{\text{tot}}$, of a chemical system. (a) The local atomic environment of every atom $i$, consisting of its type (e.g. C, H, O, ...) and information about the relative positions $\mathbf{r}_j$ and nuclear charges $Z_j$ of all neighboring atoms $j$ inside the cutoff sphere (indicated by a red circle), are encoded in a fixed-size numeric descriptor vector $\mathbf{c}_i$. (b) Since the descriptor is rotationally, translationally and permutationally invariant, all symmetry equivalent atoms are encoded in the same way. (c) The descriptor vector $\mathbf{c}_i$ is supplied to a NN, which (d) outputs an atomic energy contribution $E_i$. Finally, the individual contributions are (e) accumulated to give $E_{\text{tot}} = \sum_i E_i$. Since addition is commutative, $E_{\text{tot}}$ is automatically invariant with respect to atom permutations.

energy decomposition scheme[82–84] needs to be imposed.

## 2.1 Atomic descriptor

Individual atoms and their local environment are represented by a descriptor, which needs to encode all information relevant to predicting its atomic energy contribution (relative positions and species of neighboring atoms). Further, due to the way feed-forward NNs are designed (see section 2.2), the descriptor must be of fixed size, no matter how many atoms are present. Finally, it is advantageous if the descriptor is invariant with respect to transformations which do not alter the energy of the system. This way, translational invariance, rotational invariance and invariance with respect to permutation of equivalent atoms need not be learned explicitly by the neural network.

In this work, the atomic descriptor consists of two parts: one part encoding the atomic species (C, H, O, . . . ) and a second part which encodes the local environment up to a cutoff radius $R$. Note that an atomic descriptor that encodes species and environment separately has been proposed previously.[68] There are several reasons for introducing a cutoff. First, the energy prediction scales linearly with respect to the number of atoms present in the system of interest. Second, while the network can be trained on rather small systems, it can then be applied to much larger systems, because locally, atomic environments of small and large systems are equivalent. Finally, it is a valid assumption that most (but not all) chemical interactions, which are relevant to the energy of the system, such as bonding, are inherently short-ranged. Methods to correct for long-range interactions are well-known in the literature[60,66,71,72] and are discussed in section 3. Hence the descriptor used here combines computationally advantageous aspects with a design based on physical/chemical principles.

**Species descriptor.** In principle, the atomic species could be encoded by a single number, either by an integer identifier (e.g. H$=1$, C$=2$, N$=3$, . . . ) or by the nuclear charge $Z$ (e.g.

H= 1, C= 6, N= 7, ...). However, this introduces an ordinal relationship (e.g. H < C < N) between different atomic species, which can be detrimental to the network performance. Since neural networks are a purely numerical algorithm, ordinal relations in inputs directly correlate with the network response, which is not meaningful for atomic species. Alternatively, a one-hot[85] encoding (e.g. H = $[1\,0\,0\cdots]$, C = $[0\,1\,0\cdots]$, N = $[0\,0\,1\cdots]$) would be possible. However, two potential disadvantages of a one-hot encoding are that 1) the dimensionality of the encoding vector must necessarily be equal to the cardinality of the set of atomic species present in the data and 2) all encodings are equidistant by construction. Since, it is intuitive to expect e.g. atomic species from the same group in the periodic table to behave similar to one another, an optimal encoding should be able to directly represent these similarities.

For these reasons, the atomic species are rather encoded by *embeddings*. An embedding is a mapping from a discrete object $i$ to a vector of real numbers $\mathbf{v}_i \in \mathbb{R}^D$, where $D$ is the dimensionality of the embeddings. For example, word embeddings[86] find wide spread use in the field of natural language processing. Here, words are mapped to a comparatively low-dimensional vector space, such that semantically similar words (e.g. "red", "green", and "blue" or "king", "monarch" and "emperor") appear close to each other ($||\mathbf{v}_{\text{red}} - \mathbf{v}_{\text{blue}}|| < ||\mathbf{v}_{\text{red}} - \mathbf{v}_{\text{king}}||$). During the training process of the NN, the entries of the embedding vectors $\mathbf{v}_i$ are free parameters, such that meaningful embeddings are directly learned from data. In this work, the dimensionality $D$ of the embeddings is set equal to the number $N_g$ of distinct groups (columns) in the periodic table which are present in the reference data. For example, in the QM9 dataset, $N_g$ is 5. Note that a lower dimensionality would still allow a unique encoding of each element (albeit introducing an ordinal relation in the extreme case of $D = 1$). However, elements from the same group in the periodic table are expected to have similar properties and choosing $D = N_g$ principally allows to encode every distinct group in orthogonal directions, thus avoiding ordinal relations between species. For more details on the concept of embeddings, the reader is referred to the literature.[87]

**Environment descriptor.** All information about the local environment of a given atom $i$ up to a cutoff radius $R$ is contained in the neighborhood density function $\rho_i$ given by

$$\rho_i(\mathbf{r}) = \sum_{j,\|\mathbf{r}_j\|\leq R} Z_j \delta(\|\mathbf{r} - \mathbf{r}_j\|) \tag{1}$$

where the position $\mathbf{r} = (x, y, z)^T \in \mathbb{R}^3$ is relative to atom $i$, $Z_j$ and $\mathbf{r}_j$ are nuclear charge and relative position of neighboring atom $j$, $\delta$ is the Dirac delta function, and the sum runs over all atoms $j$ closer than $R$. The concept of a neighborhood density function has been used previously in the derivation of the SOAP similarity kernel.[69] Note that the use of relative positions $\|\mathbf{r} - \mathbf{r}_j\|$ makes $\rho_i$ translationally invariant and the commutativity of addition ensures permutational invariance. By construction, $\rho_i$ is zero everywhere except for positions $\mathbf{r}_j$ of neighboring atoms $j$, where the function value encodes the atomic species of $j$ by its nuclear charge. Thus, $\rho_i$ completely describes the local atomic environment of atom $i$ up to a distance $R$.

In order to obtain a fixed length input $\mathbf{x}_{\text{in}}$ for use in a feed-forward layer, $\rho_i$ is expanded into a basis set of fixed dimension

$$\rho_i(\mathbf{r}) \approx \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} \sum_{m=-l}^{l} c_{klm} \psi_{klm}(\mathbf{r}) \tag{2}$$

with expansion coefficients $c_{klm}$ and basis functions $\psi_{klm}(\mathbf{r}) = g_k(r; R) Y_{lm}(\theta, \phi)$, where $g_k(r; R)$ (with $k \in [0, K-1]$) are radial basis functions and $Y_{lm}(\theta, \phi)$ are spherical harmonics (with $l \in [0, L-1]$). The Zernike descriptor[67] also relies on a basis set expansion, but uses different basis functions. $K$ and $L$ define the maximum degree of the radial and angular parts of the expansion and $R$ the cutoff radius, respectively. In order to be consistent with the commonly used notation of spherical harmonics, the Cartesian coordinate vector $\mathbf{r}$ is transformed[88] to spherical coordinates.

$$\begin{aligned} r &= \|\mathbf{r}\| = \sqrt{x^2+y^2+z^2} \\ \theta &= \operatorname{arctan2}(y,x) \\ \phi &= \arccos\left(\frac{z}{\sqrt{x^2+y^2+z^2}}\right) \end{aligned} \tag{3}$$

Many different choices for the radial basis functions $g_k(r;R)$ are possible. Here

$$g_k(r;R) = s(r;R) \cdot \exp\left(-\frac{K^2}{R^2}\left(r-(k-1)\frac{R}{K}\right)^2\right) \tag{4}$$

is chosen which ensures that basis functions are evenly spaced inside the cutoff sphere. Due to the cutoff function $s(r;R)$, $g_k(r;R)$ is zero whenever $r > R$. Choosing

$$s(r;R) = \begin{cases} 1 & \text{if } r \le r_s \\ 1 - 6\left(\frac{r-r_s}{R-r_s}\right)^5 + 15\left(\frac{r-r_s}{R-r_s}\right)^4 - 10\left(\frac{r-r_s}{R-r_s}\right)^3 & \text{if } r_s < r < R \\ 0 & \text{if } r \ge R \end{cases} \tag{5}$$

as cutoff function, with $r_s = R - \frac{R}{K}$, ensures that $g_k(r;R)$ has smooth first and second derivatives, such that no numerical artifacts are introduced when an atom enters or leaves the cutoff-sphere, while leaving the Gaussian part of $g_k(r;R)$ largely unaffected (see Figure S1). The cutoff function $s(r;R)$ is a smooth approximation to the step function and influences the value of $g_k(r;R)$ only when $r > r_s$. Although it would be possible to use a non-sigmoid cutoff function that starts decaying as soon as $r > 0$, this would lead to largely different numerical influences of $g_k(r;R)$ on the network predictions depending on the value of $k$, therefore effectively introducing an *a priori* distance-based weighting. In contrast, the present choice of $s(r;R)$ allows that the NN learns to weigh the influence of different distances in a data-driven manner.

As long as $K$ and $L$ are sufficiently large, the information stored in the coefficients $c_{klm}$ is comparable to that encoded in $\rho_i$. Note that for predicting energies, some loss of information is not problematic as long as the resulting descriptor can distinguish different environments sufficiently well.

The expansion coefficients $c_{klm}$ for a general function $f(\mathbf{r})$ can be obtained from projecting $c_{klm} = \int f(\mathbf{r})\psi_{klm}(\mathbf{r})\mathrm{d}\mathbf{r}$. Fortunately, it is not necessary to calculate an integral to obtain the expansion coefficients for the neighborhood density function. Since $\rho_i(\mathbf{r})$ is the sum of $\delta$ functions (Eq. 1), the coefficients are efficiently obtained by summation

$$c_{klm} = \int \rho_i(\mathbf{r})\psi_{klm}(\mathbf{r})\mathrm{d}\mathbf{r} = \sum_{\|\mathbf{r}_j\| \leq R} Z_j \psi_{klm}(\mathbf{r}_j) \tag{6}$$

Note that the values of the coefficients $c_{klm}$ still depend on the orientation of the chosen reference coordinate system, because the values of the $2l+1$ spherical harmonics for a particular $l$ are orientation dependent. Fortunately, the $2l+1$ coefficients for given combination of $k$ and $l$ can be combined to a rotationally invariant quantity $a_{kl}$ according to (Eq. 7).

$$a_{kl} = \left( \frac{4\pi}{2l+1} \sum_{m=-l}^{m=l} (-1)^m c_{klm} c_{kl-m} \right)^{\frac{1}{2}} \tag{7}$$

In total, there are $K \cdot L$ different $a_{kl}$ values, which are concatenated to the atom embedding vector $\mathbf{v}$ of dimensionality $N_g$ to form the descriptor vector $\mathbf{c}$. Because $a_{kl}$ has continuous first derivatives with respect to the atom coordinates, derivatives necessary for e.g. force calculations are easily obtained by the chain rule. Note that because a single vector $\mathbf{c} = \mathbf{x}_{\text{in}} \in \mathbb{R}^{N_g + K \cdot L}$ is supplied to the NN, it is not able to distinguish between species and environment descriptor. In this work, $K = 7$, $L = 7$ and $R = 3$ Å are chosen for all datasets. Section S1.1 details how the values of $K$, $L$ and $R$ were chosen and how they influence the predictive accuracy of the NN.

## 2.2 Neural Network

A feed-forward NN consists of an input layer connected to one or multiple hidden layers and an output layer. Every layer can be considered as a function which takes an $n_{\text{in}}$-dimensional input vector $\mathbf{x}$ and transforms it to an $n_{\text{out}}$-dimensional output vector $\mathbf{y}$. For most NNs, the transformation in each layer can be written as

$$\mathbf{y} = \boldsymbol{\phi}(\mathbf{x}\mathbf{W} + \mathbf{b}) \tag{8}$$

where $\mathbf{W}$ is an $n_{\text{in}} \times n_{\text{out}}$ weight matrix, $\mathbf{b}$ is an $n_{\text{out}}$-dimensional bias vector, and $\phi(x)$ is the activation function. For simplicity, the shorthand notation $\boldsymbol{\phi}(\mathbf{x})$ is used, which symbolizes element-wise application of $\phi(x)$ to $\mathbf{x}$ (performed independently on each vector entry). All entries of the weight matrix $\mathbf{W}$ and the bias vector $\mathbf{b}$ are free parameters, which are initialized randomly and optimized when the network is trained.

The output $\mathbf{y}$ of each layer is the input $\mathbf{x}$ to the next successive layer until the output layer is reached. Usually, the output layer uses the identity function as activation function and its output $\mathbf{y}_{\text{out}}$ is the prediction of the neural network (it is possible to predict more than one quantity at once using the same network). The input layer applies no transformation to its input data $\mathbf{x}_{\text{in}}$ at all (the activation function is the identity function, $\mathbf{W}$ is the identity matrix and the bias vector contains only zeros) and is only used to provide data for the first hidden layer. The complete NN can therefore be written as a nested version of Eq. 8 with different weight matrices $\mathbf{W}_i$, bias vectors $\mathbf{b}_i$, and activation functions $\phi_i(x)$ for each layer

$i$. For example, a NN with two hidden layers can be written as

$$\mathbf{y}_{\text{out}} = \boldsymbol{\phi}_{\text{out}}(\boldsymbol{\phi}_2(\boldsymbol{\phi}_1(\mathbf{x}_{\text{in}}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2)\mathbf{W}_{\text{out}} + \mathbf{b}_{\text{out}}) \tag{9}$$

Note that it is not necessary to symbolically differentiate an expression such as Eq. 9 if derivatives of $\mathbf{y}_{\text{out}}$ with respect to $\mathbf{x}_{\text{in}}$ (or any weight or bias parameter) are required. Instead, analytical derivatives are efficiently calculated using automatic differentiation.[89] The network architecture can be controlled by choosing different numbers of hidden layers and nodes ("neurons") in each hidden layer (specified by $n_{\text{out}}$) and the choice of the activation function (usually, the same activation function is used for all hidden layers). Commonly used activation functions are either sigmoidal functions ($\tanh(x)$ or $(1+e^{-x})^{-1}$) or "rectifier"-like functions ($\max(0,x)$ or $\ln(1+e^{x})$).[90,91] Note that $n_{\text{out}}$ is the only hyperparameter for choosing the size of the weight matrix and bias vector of each hidden layer, as $n_{\text{in}}$ is determined by the previous layer's $n_{\text{out}}$, whereas the dimensionalities of $\mathbf{y}_{\text{out}}$ and $\mathbf{x}_{\text{in}}$ are dictated by the problem at hand.

In the present work, square unit augmented layers[92] given by

$$\mathbf{y} = \boldsymbol{\phi}(\mathbf{x}\mathbf{W1} + \mathbf{x}^2\mathbf{W2} + \mathbf{b}) \tag{10}$$

are used to construct the NN instead of ordinary layers (see Eq. 8). Here, $\mathbf{x}^2$ is shorthand notation for the element-wise square of $\mathbf{x}$. The independent weight matrices $\mathbf{W1}$ and $\mathbf{W2}$ are of size $n_{\text{in}} \times n_{\text{out}}$ and $\mathbf{b}$ and $\boldsymbol{\phi}$ are bias vector and activation function, respectively (see Eq. 8). The reason for using square unit augmented layers is that properties reminiscent of radial basis function networks[93–95] can be included at little additional computational expense,[92] provided that a sigmoidal activation function is used (see Figure S2 for an illustration).

The activation function for the hidden layers is $\phi(x) = s \cdot \operatorname{arcsinh}(x)$ where $s = 1.25673480$

ensures that $\phi(x)$ has self-normalizing properties[96] (activations converge automatically to zero mean and unit variance), similar to the recently proposed SELU[96] function. For the output layer, the identity function is used. In the present work $\text{arcsinh}(x)$ was found to give superior results compared to more commonly used activation functions such as $\tanh(x)$. One possible reason for the improved performance is that the function does not saturate for large or small values of $x$ (see Figure S2), which alleviates the vanishing gradient problem[97] and helps to improve learning.

In summary, the energy prediction consists of the following steps (see also Figure 1): 1) The descriptor $\mathbf{c}_i$ for a specific atom $i$ with nuclear charge $Z$ is generated by concatenating the embedding vector $\mathbf{v}_Z$ with the environment descriptor generated from the neighborhood density (Eq. 1) of atom $i$ (see section 2.1). 2) The descriptor $\mathbf{c}_i$ is used as input $\mathbf{x}_{\text{in}}$ for a NN, which outputs the atomic energy contribution $E_i$. All NNs used in this work consist of two hidden square unit augmented layers (Eq. 10) (see section 2.2) with 100 and 50 nodes each. 3) Steps 1 and 2 are repeated for every atom $i$ and the contributions $E_i$ are summed to give the total energy $E_{\text{tot}}$.

## 2.3 Training

NNs are trained to predict energies on the QM9 dataset,[77] several MD datasets[81] and a dataset for H-transfer in malonaldehyde. The QM9 dataset forms a subset of the GDB-17 database[98] and contains 133,885 molecules consisting of H, C, N, O and F with up to 29 atoms, including up to 9 heavy atoms. The range of energies spans several thousand kcal mol$^{-1}$. All properties in the QM9 dataset were calculated at the B3LYP/6-31G(2df,p) level of theory.[77] The MD datasets consist of *ab initio* MD trajectories for benzene, uracil, naphthalene, aspirin, salicylic acid, malonaldehyde, ethanol and toluene calculated at the PBE + vdW-TS[99,100] level of theory. They range in size from 150,000 to nearly 1,000,000 conformational geometries.[81] The H-transfer dataset for malonaldehyde was generated by sampling 250,000 geometries from a 5 ns MD trajectory run at 750 K using CHARMM[101] and a molecular mechanics with proton transfer(MMPT)-based reactive force field.[42,102] These simulation conditions lead to ready hydrogen/proton transfer and constitute a set of reactive geometries. The energy for each geometry was calculated at the MP2/6-311++G(d,p) level of theory using Gaussian09[103] and is used as reference.

Prior to training, each dataset is split into three parts: the training set, the validation set and the test set. During training, the squared error per atom (SEpA)

$$\mathrm{SEpA} = \frac{1}{N}\left(E_{\mathrm{ref}} - \sum_{i=1}^{N} E_i\right)^2 \tag{11}$$

is minimized via Adam optimization in minibatches[104] of ten reference structures, using a learning rate of $10^{-4}$. $E_{\mathrm{ref}}$ is the reference energy of a structure from the training set, and $E_i$ are the predicted atomic contributions of the $N$ atoms of the reference structure. During one so-called epoch of training, the network trains once on each datum in the training set. After each training epoch, the mean SEpA is also calculated for the structures in the validation set. Every network is trained between 5500 to 10,000 epochs and the model which performs best on the validation set is selected to predict the test set. As such, although the validation set is not directly used in training, it indirectly influences which model is selected. This method is also known as early stopping and is frequently used to prevent overfitting.[92] Since the test set is not used at all during the training process, the mean absolute error (MAE) and root mean squared error (RMSE) of predictions on the test set indicate how well the model generalizes to unknown data.

In order to speed up the training process and to improve convergence, all inputs (apart from embeddings) to the network are transformed to their z-score[105] according to mean and standard deviation of the respective inputs in the training set. This ensures that the numerical range of input values is close to the regions where the activation function is most responsive. Note that all numbers needed for calculating the z-scores are constants that only depend on the chosen training set and can be considered to be part of the descriptor. The transformation to z-scores or similar normalization methods have only numerical reasons and are standard practice when working with NNs.[92]

Similarly, instead of directly interpreting the output $y_{\text{out}}$ of the NN as atomic contribution to the energy, $E_i = \sigma \cdot y_{\text{out}} + \mu$ is used instead, where $\sigma$ and $\mu$ are additional scale and shift parameters that are optimized during training. However, instead of initializing them randomly like the other trainable parameters, they are initialized according to the standard deviation ($\sigma$) and mean ($\mu$) of the per-atom average of the reference energies in the training set. Note that introducing $\sigma$ and $\mu$ is redundant, because both, scaling and shift operations, can already be equivalently expressed through the parameters in $\mathbf{W}_{\text{out}}$ and $\mathbf{b}_{\text{out}}$ of the output layer. However, networks are found to converge faster when $\sigma$ and $\mu$ are introduced, because a larger learning rate can be used due to the network predictions starting with the correct range of values. After training is finished, it is possible to incorporate $\sigma$ and $\mu$ directly into $\mathbf{W}_{\text{out}}$ and $\mathbf{b}_{\text{out}}$ to save the additional computational step required by introducing $E_i = \sigma \cdot y_{\text{out}} + \mu$ instead of simply choosing $E_i = y_{\text{out}}$.

NNs are trained with Tensorflow[106] using training set sizes of 1k, 2.5k, 5k, 10k, 25k, 35k, 50k, 75k and 100k for the QM9 dataset and training set sizes of 25k, 50k and 100k for the MD and H-transfer datasets. In all cases, 2k additional structures are used as validation set, whereas the remaining structures constitute the test set. For every training set size in the QM9 dataset, five different NNs are trained based on a different, randomly chosen training, validation and test set. This provides a means to obtain statistics on their performance.

Further, to investigate whether the predictions of a NN also scale to larger systems, a single network is trained on the QM9 dataset only on reference structures that contain 15 atoms or less (26,328 structures). Out of the remaining structures, 2k are reserved as validation set during training and the generalization error is estimated by predicting the energies of all other structures in the QM9 dataset with more than 15 atoms.

# 3 Results and Discussion

## 3.1 Atomic energies

Since the NNs are trained to decompose energies of a system into atomic contributions, it is instructive to visualize the "energy spectrum" for each atomic species in the QM9 dataset (Figure 2).

The spectra are non-uniform and contain multiple peaks at well defined energies. Intuitively one would associate different peaks to different clusters ("types") of atoms, where atoms in the same cluster are similar in energy $E_i$ due to similar atomic environments. In order to verify this hypothesis, atoms with similar environments are clustered based on chemical graphs,[108] where nodes correspond to atoms and edges represent bonds. Different atoms are distinguished by a string (similar to a SMILES string[109]), which is obtained by concatenating labels for all nodes encountered in a depth-first[110] tree traversal of the chemical graph up to depth two, starting from the atom of interest. The node labels consist of atomic species and the number of edges to other nodes. Atoms with identical strings are assigned to the same cluster. A more detailed description of the clustering method is available in section S4 of the SI.

In total, the QM9[77] dataset contains 1,230,122 H atoms, 846,557 C atoms, 139,764 N atoms,

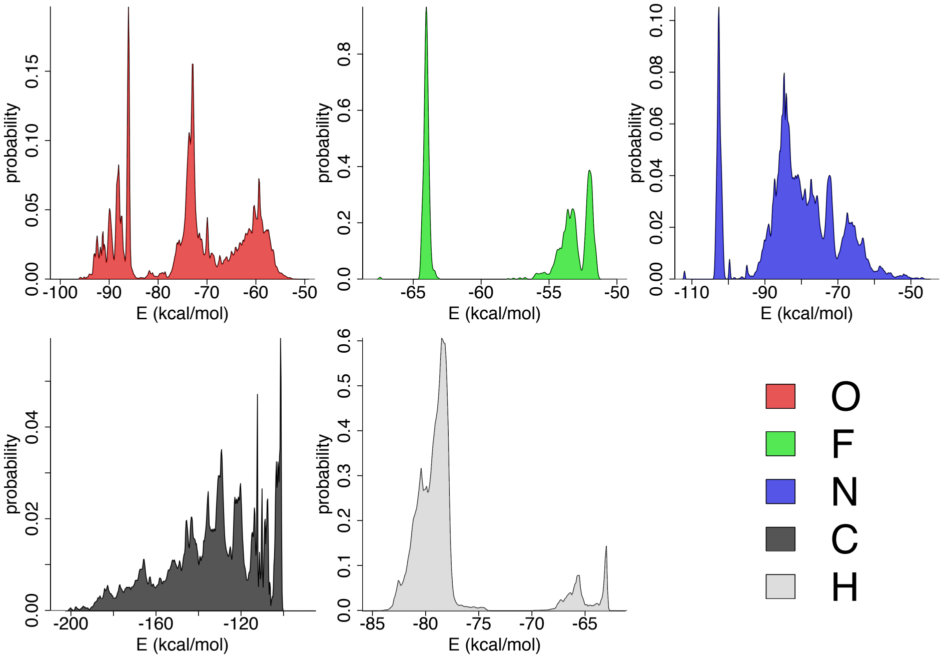


Figure 2: "Spectra" of atomic energies in the QM9 dataset for different species (relative to the energy of a free atom). In order to obtain the spectra, the atomic energy predictions of all five NNs trained on 100k structures were averaged and the curves are obtained by kernel density estimation with the Sheather-Jones bandwidth selection method.[107] Figures S7, S8, S9, S10 and S11 show the respective unaveraged results. The atomic energies of C atoms span the widest range ($> 100$ kcal mol$^{-1}$), followed by N ($> 60$ kcal mol$^{-1}$), O ($> 40$ kcal mol$^{-1}$), H ($> 20$ kcal mol$^{-1}$) and F ($> 15$ kcal mol$^{-1}$).

187,996 O atoms and 3314 F atoms which reduces to 168 (H), 34,647 (C), 4271 (N), 1130 (O), and 22 (F) after clustering (for detailed results, see section S4). The large number of different clusters is not surprising, considering the vast number of theoretically possible combinations for constructing bonding graphs of depth two, given five different atomic species and diverse possible bonding patterns for each of them (see Table S1 for an illustration of the exponential growth of possible combinations when traversing the bonding graph). Interestingly, however, most atoms can be assigned to just a few clusters (see Figure S5). For example, more than half of all C atoms belong to the 331 most common C-atom clusters.

Since only graph-based information (but no geometric information such as distances and angles) is considered in the clustering approach, it is not evident that atoms belonging to the same cluster are energetically similar. As a qualitative test for how meaningful the clustering is, the cluster statistics (mean and variance of atomic energies for each cluster) from the raw data is considered (see Figure 2). For this, every cluster is represented by a Gaussian distribution with mean and variance equal to the corresponding cluster statistics, and normalized according to the atom count. Even though assuming a Gaussian distribution is a crude approximation, the sum of all Gaussians (see Figure S6) closely resembles Figure 2, so the graph-based clustering approach is considered to be meaningful.

In order to interpret the data, chemical similarities between different clusters are analyzed and they are summarized based on functional groups into different atom types. Apart from allowing interpretation of the network predictions, the energies of different atom types can be tabulated and used for a rapid estimate of the energy of a molecule given only its chemical structure, similar to how NMR-chemical shifts can be estimated.[111] Table 1 lists atomic energies (relative to a free atom) of functionally different C atom types.

Table 1: Environment-dependent atomic energies of selected **C** atom types (mean plus or minus one standard deviation).

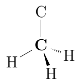
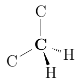
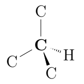
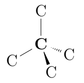
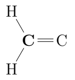
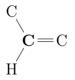
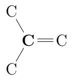
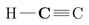
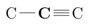
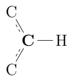
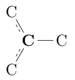
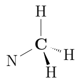
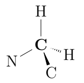
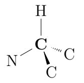
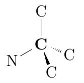
| type | diagram | $\overline{E}$ (kcal/mol) |
|---|---|---|
| hydrocarbyls | | |
| primary alkyl |  | $-101.8 \pm 0.7$ |
| secondary alkyl |  | $-114.3 \pm 4.8$ |
| tertiary alkyl |  | $-129.5 \pm 6.3$ |
| quaternary alkyl |  | $-151.3 \pm 5.9$ |
| primary alkenyl |  | $-110.0 \pm 6.2$ |
| secondary alkenyl |  | $-128.6 \pm 2.1$ |
| tertiary alkenyl |  | $-150.2 \pm 8.3$ |
| primary alkynyl |  | $-112.5 \pm 1.0$ |
| secondary alkynyl |  | $-137.1 \pm 3.0$ |
| secondary conjugated alkenyl |  | $-134.0 \pm 3.0$ |
| tertiary conjugated alkenyl |  | $-156.9 \pm 5.6$ |
| bound to nitrogen | | |
| methyl amine |  | $-108.6 \pm 0.5$ |
| primary-C amine |  | $-120.2 \pm 7.4$ |
| secondary-C amine |  | $-134.0 \pm 8.1$ |
| tertiary-C amine |  | $-156.7 \pm 7.1$ |

Table 1: Environment dependent atomic energies of selected **C** atom types. (*continued*)

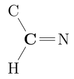
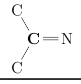
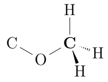
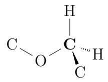
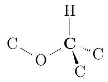
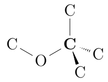
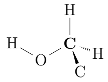
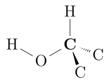
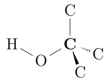
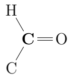
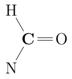
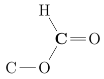
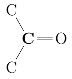
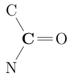
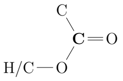
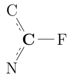
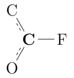
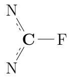
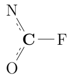
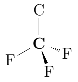
| type | diagram | $\overline{E}$ (kcal/mol) |
|---|---|---|
| nitrile | C—**C**≡N | $-144.0 \pm 1.9$ |
| primary-C imine |  | $-142.0 \pm 2.2$ |
| secondary-C imine |  | $-165.6 \pm 3.7$ |
| bound to oxygen (may also be bound to nitrogen) | | |
| methoxy |  | $-110.3 \pm 0.1$ |
| primary ether |  | $-126.8 \pm 6.0$ |
| secondary ether |  | $-141.2 \pm 8.4$ |
| tertiary ether |  | $-161.4 \pm 7.5$ |
| primary hydroxyl |  | $-130.3 \pm 1.3$ |
| secondary hydroxyl |  | $-149.3 \pm 4.0$ |
| tertiary hydroxyl |  | $-168.0 \pm 4.7$ |
| aldehyde |  | $-146.0 \pm 1.3$ |
| formyl amide |  | $-161.6 \pm 1.4$ |
| formyl ester |  | $-164.5 \pm 1.1$ |
| ketone |  | $-167.2 \pm 2.3$ |
| amide |  | $-184.1 \pm 4.0$ |
| carboxyl ester/acid |  | $-188.9 \pm 4.7$ |
| bound to fluorine | | |

Table 1: Environment dependent atomic energies of selected **C** atom types. (*continued*)

| type | diagram | $\overline{E}$ (kcal/mol) |
|---|---|---|
| "aza-conjugated" fluoro |  | $-181.1 \pm 1.3$ |
| "oxy-conjugated" fluoro |  | $-182.9 \pm 1.5$ |
| "aza-aza-conjugated" fluoro |  | $-188.0 \pm 1.3$ |
| "aza-oxy-conjugated" fluoro |  | $-192.9 \pm 1.0$ |
| fluoro methyl |  | $-200.2 \pm 0.5$ |

Several trends can be observed: For pure hydrocarbyls, C atoms with a triple bond are more stable than C atoms with double or single bonds, in accordance with the increased bond strengths. An exception are conjugated sp$^2$-hybridized C atoms, which are even more stable due to their "aromatic" nature. When bound to electronegative atoms, such as N, O and F, the stabilization energy of carbon atoms appears to be correlated with the electronegativity of the bonding partner. A physically appealing interpretation is that a large difference in electronegativity increases the ionic character of the bond and therefore increases the stabilization energy.

While such trends may be obvious to chemists, a somewhat more subtle effect can be seen in the increasing stability from primary to quaternary C-atoms. This can be explained by hyperconjugation[108] (electron density from occupied $\sigma$-bonds is donated to unoccupied orbitals, also known as the positive inductive or +I effect[112]). Such a resonance stabilization is well known for carbocations and carbon radicals, which become more stable with increasing number of neighboring alkyl groups. A related trend is found from chemical shift measurements in $^{13}$C-NMR experiments, where typical shifts increase from 15-30 ppm, to 22-45 ppm and further to 30-58 ppm when going from primary to tertiary C-atoms.[113] This is usually attributed to the increased nuclear shielding due to the additional electron density around the nucleus.

Similar observations are made for H, N, O and F atoms (see Tables S2, S3, S4 and S5). Note that some of the previously discussed trends can be reversed for the other elements. For example, instead of being stabilized by neighboring alkyl groups, O atoms typically are destabilized by the +I effect. However, this is to be expected since O atoms are already partially negatively charged due to their high electronegativity. The +I effect then leads to an amplification of this charge and therefore destabilization.

## 3.2 Errors

**QM9 dataset.**[77] Mean absolute errors (MAEs) and root mean squared errors (RMSEs) for the NNs trained with different training set sizes are summarized in Table 2 and compared with the performance of the DTNN[73] and SchNet.[78]

The NN trained on 100k reference structures predicts structures in the QM9 dataset accurately with a MAE of 0.41 kcal mol$^{-1}$ and an RMSE of 0.86 kcal mol$^{-1}$. Note that SchNet has lower errors for larger training set sizes, but is outperformed by the present approach for smaller training sets. Also, SchNet does not employ a cutoff radius $R$ and therefore uses significantly more information in its prediction. Figure 3 shows the convergence of MAE and RMSE with increasing training set size.

While MAE and RMSE are useful measures for the overall performance of a method, it can also be instructive to consider how errors are distributed. Figure 4 reveals that for all training set sizes starting from 10k, more than half of all errors are below 0.5 kcal mol$^{-1}$,

Table 2: Prediction errors for the QM9 dataset. MAE and RMSE (given in kcal mol$^{-1}$) on the test set for different training set sizes. Results for this work and refs. 73 and 78 are compared.

| | training set | MAE | RMSE |
|---|---|---|---|
| this work | 1000 | $1.85 \pm 0.09$ | $3.53 \pm 0.57$ |
| | 2500 | $1.23 \pm 0.03$ | $2.45 \pm 0.14$ |
| | 5000 | $0.95 \pm 0.01$ | $1.94 \pm 0.10$ |
| | 10,000 | $0.73 \pm 0.01$ | $1.59 \pm 0.08$ |
| | 15,000 | $0.63 \pm 0.01$ | $1.40 \pm 0.08$ |
| | 25,000 | $0.55 \pm 0.01$ | $1.22 \pm 0.07$ |
| | 35,000 | $0.50 \pm 0.00$ | $1.06 \pm 0.02$ |
| | 50,000 | $0.46 \pm 0.01$ | $0.98 \pm 0.04$ |
| | 75,000 | $0.43 \pm 0.01$ | $0.89 \pm 0.06$ |
| | 100,000 | $0.41 \pm 0.00$ | $0.86 \pm 0.14$ |
| DTNN[73] | 25,000 | $1.04 \pm 0.02$ | $1.53 \pm 0.02$ |
| | 50,000 | $0.94 \pm 0.01$ | $1.37 \pm 0.01$ |
| | 100,000 | $0.84 \pm 0.02$ | $1.21 \pm 0.02$ |
| SchNet[78] | 50,000 | 0.59 | — |
| | 100,000 | 0.34 | — |
| | 110,462 | 0.31 | — |

with most errors being as small as $< 0.1$ kcal mol$^{-1}$. However, all distributions exhibit long tails, which implies that there are rare but extreme outliers. The question remains whether reasons for the outliers can be identified.

The energies of particular structures could be difficult to predict simply because they contain rare atomic environments which are underrepresented in the training set. In order to quantify how well a structure in the test set is represented by structures in the training set, the concept of a representation number is introduced. For every structure, the relative frequency of the atom clusters (see section 3.1) in the training set are combined via a harmonic average to form the structure's representation number. Structures with a small representation number therefore contain one or several uncommon atomic environments, which the network could not necessarily learn to predict accurately from the data it was presented during training. Notable examples for such structures are very small molecules, including water, methane and fluoromethane (all part of the QM9 dataset), which contain unique atomic environments not

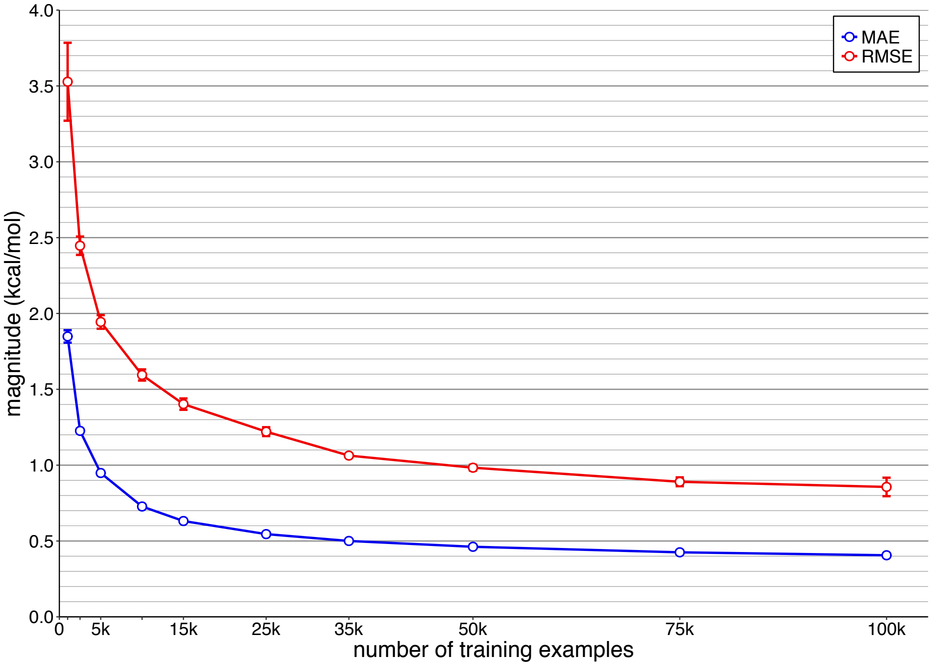


Figure 3: MAE (blue) and RMSE (red) depending on the size of the training set, averaged over five independent runs per training set size. The error bars indicate one standard deviation.

found in any other structure. For example, oxygen and hydrogen atoms in a water molecule are chemically very different to oxygen and hydrogen atoms found in other hydroxyl groups. This is highlighted by noting that the bond dissociation energy of an O-H bond in water is 119.2 kcal mol$^{-1}$, whereas for a typical O-H bond in hydroxyl groups, it is only 102.3 kcal mol$^{-1}$.[114] Similarly, the dissociation energy of C-H bond in methane is around 103.0 kcal mol$^{-1}$, compared with 113.0 kcal mol$^{-1}$ for a typical C-H bond to a primary carbon.[114] Figure S3 reveals that particularly large prediction errors occur almost exclusively for structures with low representation number. However, low representation numbers do not necessarily lead to large prediction errors. Since outliers follow the same patterns, it is possible to systematically improve the prediction capabilities of the network for structures with a low representation number by simply including appropriate reference structures in the training set and it may even be possible to use the present approach for database curation and quality tests of databases, which is essential for meaningful ML applications.

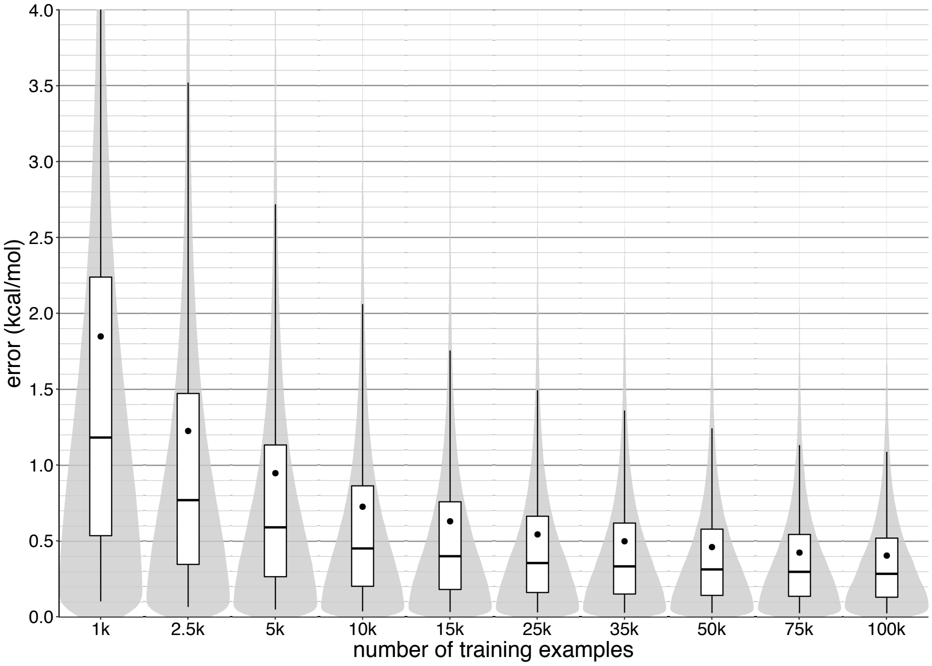


Figure 4: Normalized error distribution (grey) depending on the size of the training set (test errors from five independent runs per training set size are combined). A white box spans between the 25% and 75% quantiles, with a black horizontal line indicating the median and a black dot indicating the mean of the distribution. The whiskers mark the 5% and 95% quantiles.

While most outliers can be explained by underrepresented environments in the training data, some of the largest prediction errors are probably due to a different reason. They belong to a group of eleven molecules in the QM9 dataset for which the electronic structure calculation did not converge at all (three molecules) or only using loose convergence criteria (eight molecules).[77] Most of these structures feature unconventional chemical bonding and their electronic structure potentially has multi-reference character. Therefore, it is possible that the quantum mechanical reference energies themselves are erroneous for these structures, explaining the large prediction errors. At the very least, they seem to be particularly difficult to predict for *ab initio* methods as well.

The ability of the NN to identify problematic structures can even be advantageous to detect failures of the *ab initio* method used to obtain the reference energy and can be used to automatically identify inconsistencies in a reference database. It might turn out that the predictions by the network are closer to experiment than the reference values. Some of the difficult-to-converge structures are shown in Figure 5 along with their average prediction errors.

While the results with randomly chosen training sets are promising, it is interesting to see whether representations learned from small structures can be used to predict energies for larger structures. A NN trained on all structures in the QM9 dataset containing up to 15 atoms (26,328 structures) is able to predict structures with more than 15 atoms (107,557) with a MAE of 1.01 kcal mol$^{-1}$ and an RMSE of 1.69 kcal mol$^{-1}$. The distribution of errors is similar to the error distributions of networks trained with randomly chosen training sets, but based on all molecules (with up to 29 atoms) (see Fig. S4). This demonstrates that the learned representations are transferable and can be used to accurately predict larger structures. Nonetheless, the performance is inferior compared to a randomly chosen training sets drawn from the full data set. One possible physical explanation is that this is due to

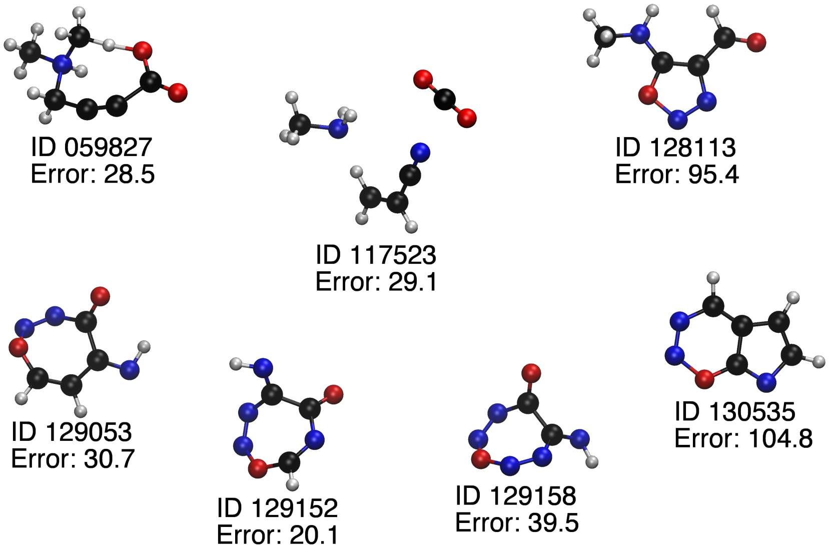


Figure 5: Structures (C = black, N = blue, O = red, H = white) with particularly large prediction errors (in kcal mol$^{-1}$) are shown along with their corresponding ID in the QM9 dataset. They all belong to a group of eleven molecules for which the reference electronic structure was difficult to converge.[77] The structures with the IDs 129158 and 117523 could not be converged at all.[77] Prediction errors are averaged across neural networks trained on 100k reference structures (only NNs that contain a given structure in the test set were considered). Note that, even though many of the structures contain a motif reminiscent of 1,2,3-oxadiazole, the presence of this motif alone can not be the cause for the large prediction errors: the QM9 dataset contains close to 1k structures with similar motifs, for which accurate predictions are possible.

the lack of an adequate description of long range interactions, which are more important for extended structures containing many atoms. These deficiencies could be addressed by explicitly including long range contributions into the prediction.

**MD datasets.**[81] MAEs and RMSEs for the NNs trained with different training set sizes are summarized in Table 3 and compared with results for gradient-domain machine learning (GDML).[81]

Table 3: Prediction errors for the MD datasets. MAE and RMSE (given in kcal mol$^{-1}$) on the test sets are given for different training set sizes. Values in brackets are results for gradient-domain machine learning (GDML).[81] Note that the GDML approach uses different reference data and training set size (see text).

| molecule | training set | MAE | | RMSE | |
|---|---|---|---|---|---|
| | 25,000 | 0.45 | | 0.61 | |
| aspirin | 50,000 | 0.34 | (0.27) | 0.44 | (0.36) |
| | 100,000 | 0.27 | | 0.35 | |
| | 25,000 | 0.11 | | 0.14 | |
| benzene | 50,000 | 0.10 | (0.07) | 0.13 | (0.09) |
| | 100,000 | 0.09 | | 0.12 | |
| | 25,000 | 0.21 | | 0.30 | |
| ethanol | 50,000 | 0.18 | (0.15) | 0.24 | (0.20) |
| | 100,000 | 0.15 | | 0.20 | |
| | 25,000 | 0.44 | | 0.60 | |
| malonaldehyde | 50,000 | 0.38 | (0.16) | 0.51 | (0.25) |
| | 100,000 | 0.32 | | 0.43 | |
| | 25,000 | 0.41 | | 0.54 | |
| naphthalene | 50,000 | 0.37 | (0.12) | 0.47 | (0.15) |
| | 100,000 | 0.32 | | 0.42 | |
| | 25,000 | 0.44 | | 0.59 | |
| salicylic acid | 50,000 | 0.37 | (0.12) | 0.48 | (0.15) |
| | 100,000 | 0.32 | | 0.42 | |
| | 25,000 | 0.45 | | 0.60 | |
| toluene | 50,000 | 0.40 | (0.12) | 0.52 | (0.16) |
| | 100,000 | 0.35 | | 0.46 | |
| | 25,000 | 0.30 | | 0.40 | |
| uracil | 50,000 | 0.24 | (0.11) | 0.31 | (0.14) |
| | 100,000 | 0.20 | | 0.26 | |

Predictions are accurate for all molecules and can be systematically improved by increasing the training set size. Even though the present approach is outperformed by GDML in some cases, it is important to keep in mind that GDML does not employ a spatial cutoff. Therefore, it is questionable whether GDML scales well to larger systems. Further, while the GDML models are trained on only 1000 structures, they use the atomic forces instead of total energies as reference data, which enhances their predictive power.[81] It has been shown previously that NNs benefit as well from including forces in their loss function (see Eq. 11).[78] Hence, it is likely that predictions from the NN could be further improved by including force information during training.

**H-transfer dataset.** MAEs and RMSEs for the NNs trained with different training set sizes for malonaldehyde are summarized in Table 4. The results show that accurate predictions are possible with rather small training set sizes and can be systematically improved by increasing the number of reference structures. Malonaldehyde has been used previously as a model reactive system in machine learning applications for bypassing the solution of the Kohn-Sham equations.[115] Figure 6 shows a 10 ps MD trajectory of malonaldehyde. Note that the NN approach automatically leads to a reactive PES. A direct comparison of the NN-learned and MP2-reference energies yields a correlation coefficient of 0.997.

Table 4: Prediction errors for the H-transfer dataset. MAE and RMSE (given in kcal mol$^{-1}$) on the test sets are given for different training set sizes.

| training set | MAE | RMSE |
|---:|---:|---:|
| 25,000 | 0.36 | 0.49 |
| 50,000 | 0.30 | 0.40 |
| 100,000 | 0.25 | 0.34 |

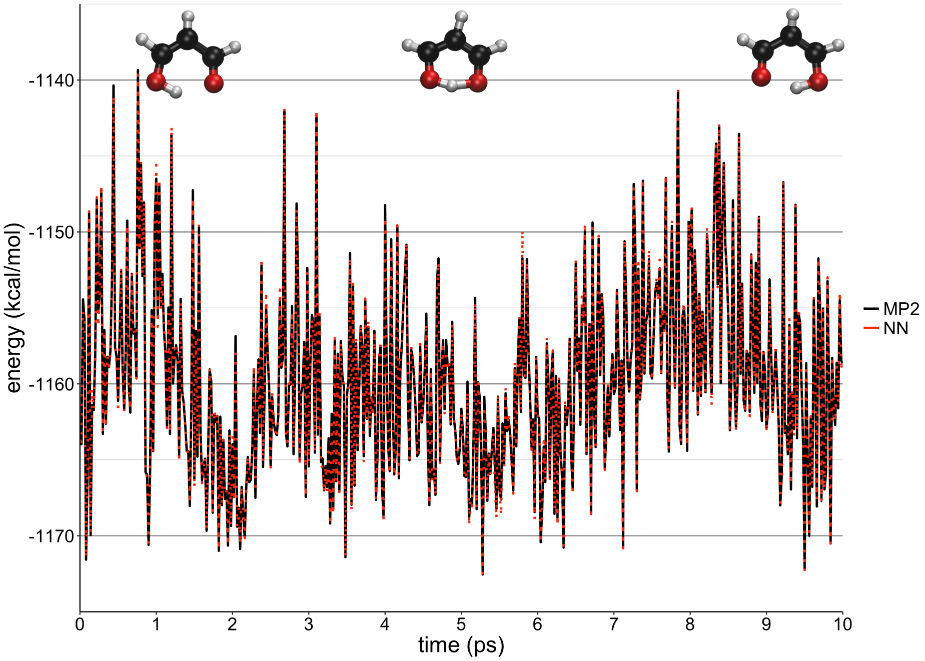


Figure 6: First 10 ps of a MD trajectory of malonaldehyde with intramolecular H-transfer. *Top panel:* Energy difference (absolute error) between MP2/6-311++G(d,p) reference energies and energies predicted by the NN trained on 100k reference structures. The error rarely exceeds 1 kcal mol$^{-1}$. *Bottom panel:* The solid black curve corresponds to the reference energies, the dotted red curvecorresponds to the energies predicted by the NN. It is able to describe transition geometries and geometries close to equilibrium structures equally well.

# 4 Discussion and Conclusion

Although the results show that accurate predictions can be obtained from training a NN with a descriptor based on encoding the chemical environment of an atom, it is useful to discuss potential problems and possible improvements to the prediction method. For example, even though introducing a cutoff radius $R$ is necessary for computational efficiency, it can limit the accuracy of the neural network. Since all atoms beyond the cutoff radius of $R = 3$ Å are ignored in the descriptor by construction, interactions extending over larger distances can not be captured by the present approach. Most interactions relevant in chemistry are sufficiently short ranged that this is not an issue, but there are important exceptions: Coulomb and dispersion interactions. These long range contributions to the total energy are especially important for the correct description of intermolecular interactions and are therefore crucial for condensed phase systems. While it is always possible to increase $R$ until the error introduced by the cutoff is negligible, this is not very efficient, as a larger number of atoms would need to be considered for the calculation of the expansion coefficients $c_{klm}$ (see Eq. 1 and Eq. 6). Further, it is likely that higher order expansion terms (see Eq. 2) are necessary to resolve differences between atomic environments for larger $R$, such that the calculation of the descriptor becomes more expensive. Fortunately, the physical laws governing Coulomb and dispersion interaction are well understood, such that it is possible to include both contributions explicitly without increasing the cutoff $R$.

For better describing Coulomb interactions, separately trained neural networks have previously been used[116] to predict environment-dependent Hirshfeld charges.[117] The electrostatic contribution $E_{\text{ele}}$ is then simply subtracted from the total energy $E_{\text{tot}}$ prior to training networks for predicting the short range contributions. The total energy can be recovered by combining electrostatic energies calculated from the predicted charges and the short range contributions. Note that only charge-charge interactions are necessary for the calculation of the electrostatic energy, as interactions between higher multipoles[118,119] decay faster and can therefore be implicitly described in the short range contributions.[60] In order to apply a similar method to the approach presented in this work, it is not necessary to introduce a second NN. Instead, the existing network could simply be trained to predict an atomic energy contribution $E_i$ and an environment dependent charge $q_i$ simultaneously, by introducing a second network output and an appropriate modification of the objective function (Eq. 11). Also, it is not necessary to rely on a charge decomposition scheme such as Hirshfeld's method[117] to obtain a reference value for $q_i$. Recently, it was shown that a NN can be trained to predict environment dependent charges such that the electrostatic moments, a true quantum mechanical observable, are reproduced.[120] This way, no arbitrary decomposition scheme needs to be imposed.

To account for long range dispersion interactions, it was shown[121] that the D3 scheme in DFT calculations proposed by Grimme[122] can be used for NNs without modification. Since the neural network is trained on DFT reference energies, the standard $C_6$ coefficients[122] for calculating the dispersion interaction can be reused. The possibility of predicting environment-dependent $C_6$ coefficients, instead of using constant values, should be pointed out. That way the dispersion correction is more flexible and can adapt to the reference data. This would require the introduction of another network output and a suitable modification of the objective function (Eq. 11), similar to the possible treatment of Coulomb interactions. Recently, it was shown that van der Waals interactions are essential for the understanding of the properties of liquid water.[123] These findings show the importance of a correct treatment of long-range dispersion when studying condensed-phase systems.

In the present work a general atomic descriptor, which is applicable to any chemical system was introduced. Using the descriptor as input, NNs trained on 100k reference structures can learn to accurately predict energies of structures in the QM9 dataset[77] across chemical space with a MAE of 0.41 kcal mol$^{-1}$. Although the performance is slightly worse than that of the

SchNet architecture[78] (MAE of 0.34 kcal mol$^{-1}$), the difference in accuracy is considered to be an acceptable trade-off for the increased computational efficiency, as the atomic descriptor developed here requires only strictly local information (due to the introduction of a cutoff radius $R$) and the network architecture is much simpler. This allows efficient calculation of thousands of atomic contributions in parallel, which is an advantage in the context of a large molecular dynamics simulation. For smaller training set sizes (e.g. 50k reference structures), the method proposed in this work outperforms SchNet (Table 2). As such, fewer reference calculations are needed to obtain chemical accuracy.

Since the QM9 dataset contains exclusively equilibrium structures it is only suited to assess transferability across chemical space. In order to demonstrate the predictive power of a NN across configurational space, the same method was also applied to data sampled from MD simulations. Using 100,000 reference structures, MAEs between 0.09 and 0.35 kcal mol$^{-1}$ were obtained (see Table 3). Finally, it was also demonstrated that this network can be used to describe chemical reactions (here proton transfer), provided that appropriate reference structures are included in the training set. The NN is able to describe intramolecular H-transfer in malonaldehyde with a similar quality as high-level *ab initio* methods (Table 4).

The present approach is particularly suitable to evaluate accurate energies. In principle, it also allows to efficiently evaluate forces as is required in molecular dynamics simulations. In addition, the method automatically leads to a reactive PES (provided that appropriate structures around the transition state are contained in the training set), as no notion of chemical bonds is introduced in the construction of the atomic descriptor. In the present work it was demonstrated that NNs trained on systems containing few atoms are transferable to larger systems which facilitates the possibility to train networks using very accurate *ab initio* reference energies. While they are typically slower than empirical force fields by one to two orders of magnitude, the energy prediction is several orders of magnitudes faster than *ab initio* methods (the energy prediction of a system with 17 atoms takes $< 1$ ms on on a desktop computer equipped with an Intel Xeon Processor E3-1275 at 3.40 GHz) and scales linearly with respect to the number of atoms. On the same machine, training the NN takes approximately three weeks and only needs to be performed once. Depending on system size and level of theory, this is approximately the same time scale as a single *ab initio* calculation. While FFs are still undisputedly the fastest approximate method, NNs promise huge potential speedups and it might be feasible to combine the two to a hybrid approach similar to QM/MM methods.

The atomic energy contributions predicted by the network are chemically intuitive and may offer new insights. For example, they can be used as a guideline for designing novel types of empirical force fields through atom typing based on quantitative information instead of chemical intuition. Finally, it is possible to systematically improve the predictions of the neural network by simply adding new reference data to the training set. As such, several properties of an “ideal PES” as put forward in the introduction are fulfilled by the present approach.

In order to use the present approach in MD simulations in a similar manner to FFs, an appropriate reference dataset is necessary to train the NN. Ideally, this dataset should contain a multitude of different chemical structures, representative of both, equilibrium and non-equilibrium geometries. For a meaningful description of reactions, transition state geometries need to be included as well. Future work will focus on using the present technology in conventional and reactive MD simulations together with a physically motivated treatment of long range contributions to the energy. This is necessary to correctly describe the intermolecular interactions governing the dynamics in condensed phase simulations.

# Acknowledgments

The authors acknowledge financial support from the Swiss National Science Foundation (NCCR-MUST and grant 200021-7117810) and the University of Basel. OU acknowledges fruitful discussions with Thomas Brandl, Lorenzo D. Bizzini, Rajesh Mannancherry and Patrick Zwick concerning the interpretation of trends among atomic energy contributions. The authors express their gratitude to Zhen-Hao Xu for generating the data set for malonaldehyde.

# Supporting Information

Supporting information for this article is available online.

# Notes and References

(1) Dirac, P. *Proc. R. Soc. Lond. Ser. A* **1929**, *123*, 714–733.

(2) Parr, R. G. *Horizons of Quantum Chemistry*; Springer, 1980; pp 5–15.

(3) Pople, J. *Angew. Chem. Int. Ed.* **1999**, *38*, 1894–1902.

(4) Franke, R.; Nielson, G. *Int. J. Numer. Meth. Eng.* **1980**, *15*, 1691–1704.

(5) Nguyen, K. A.; Rossi, I.; Truhlar, D. G. *J. Chem. Phys.* **1995**, *103*, 5522–5530.

(6) Bettens, R. P.; Collins, M. A. *J. Chem. Phys.* **1999**, *111*, 816–826.

(7) Lancaster, P.; Salkauskas, K. *Math. Comput.* **1981**, *37*, 141–158.

(8) Ischtwan, J.; Collins, M. A. *J. Chem. Phys.* **1994**, *100*, 8080–8088.

(9) Dawes, R.; Thompson, D. L.; Wagner, A. F.; Minkoff, M. *J. Chem. Phys.* **2008**, *128*, 084107.

(10) Cassam-Chenaï, P.; Patras, F. *J. Math. Chem.* **2008**, *44*, 938–966.

(11) Braams, B. J.; Bowman, J. M. *Int. Rev. Phys. Chem.* **2009**, *28*, 577–606.

(12) Paukku, Y.; Yang, K. R.; Varga, Z.; Truhlar, D. G. *J. Chem. Phys.* **2013**, *139*, 044309.

(13) Ho, T.-S.; Rabitz, H. *J. Chem. Phys.* **1996**, *104*, 2584–2597.

(14) Hollebeek, T.; Ho, T.-S.; Rabitz, H. *J. Chem. Phys.* **1997**, *106*, 7223–7227.

(15) Hollebeek, T.; Ho, T.-S.; Rabitz, H. *Annu. Rev. Phys. Chem.* **1999**, *50*, 537–570.

(16) Unke, O. T.; Meuwly, M. *J. Chem. Inf. and Mod.* **2017**, *57*, 1923–1931.

(17) Hédin, F.; El Hage, K.; Meuwly, M. *J. Chem. Inf. and Mod.* **2016**, *56*, 1479–1489.

(18) Karplus, M. *Angew. Chem. Int. Ed.* **2014**, *53*, 9992–10005.

(19) Warshel, A. *Angew. Chem. Int. Ed.* **2014**, *53*, 10020–10031.

(20) Feig, M.; Yu, I.; Wang, P.-h.; Nawrocki, G.; Sugita, Y. *J. Phys. Chem. B* **2017**, *121*, 8009–8025.

(21) Brooks, B. R.; Bruccoleri, R. E.; Olafson, B. D.; States, D. J.; Swaminathan, S. a.; Karplus, M. *J. Comput. Chem.* **1983**, *4*, 187–217.

(22) Allinger, N. L.; Yuh, Y. H.; Lii, J. H. *J. Am. Chem. Soc.* **1989**, *111*, 8551–8566.

(23) Mayo, S. L.; Olafson, B. D.; Goddard, W. A. *J. Phys. Chem.* **1990**, *94*, 8897–8909.

(24) Rappé, A. K.; Casewit, C. J.; Colwell, K.; Goddard Iii, W.; Skiff, W. *J. Am. Chem. Soc.* **1992**, *114*, 10024–10035.

(25) Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A. *J. Am. Chem. Soc.* **1995**, *117*, 5179–5197.

(26) Mackerell, A. D. *J. Comput. Chem.* **2004**, *25*, 1584–1604.

(27) Root, D. M.; Landis, C. R.; Cleveland, T. *J. Am. Chem. Soc.* **1993**, *115*, 4201–4209.

(28) Deeth, R. J. *Chem. Commun.* **2006**, 2551–2553.

(29) Gresh, N.; Cisneros, G. A.; Darden, T. A.; Piquemal, J.-P. *J. Chem. Theory. Comput.* **2007**, *3*, 1960–1986.

(30) Tubert-Brohman, I.; Schmid, M.; Meuwly, M. *J. Chem. Theory. Comput.* **2009**, *5*, 530–539.

(31) Baskes, M. *Phys. Rev. B* **1992**, *46*, 2727.

(32) Baskes, M.; Johnson, R. *Model. Simul. Mater. Sci. Eng.* **1994**, *2*, 147.

(33) Tersoff, J. *Phys. Rev. Lett.* **1986**, *56*, 632.

(34) Tersoff, J. *Phys. Rev. B* **1988**, *37*, 6991.

(35) Brenner, D. W.; Garrison, B. J. *Phys. Rev. B* **1986**, *34*, 1304.

(36) Van Duin, A. C.; Dasgupta, S.; Lorant, F.; Goddard, W. A. *J. Phys. Chem. A* **2001**, *105*, 9396–9409.

(37) Danielsson, J.; Meuwly, M. *J. Chem. Theory. Comput.* **2008**, *4*, 1083.

(38) Nagy, T.; Yosa Reyes, J.; Meuwly, M. *J. Chem. Theory. Comput.* **2014**, *10*, 1366–1375.

(39) Hartke, B.; Grimme, S. *Phys Chem Chem Phys.* **2015**, *17*, 16715–16718.

(40) Yosa Reyes, J.; Brickel, S.; Unke, O. T.; Meuwly, M. *Phys Chem Chem Phys.* **2016**, *18*, 6780–6788.

(41) Brickel, S.; Meuwly, M. *J. Phys. Chem. A* **2017**, *121*, 5079–5087.

(42) Lammers, S.; Lutz, S.; Meuwly, M. *J. Comput. Chem* **2008**, *29*, 1048–1063.

(43) Samuel, A. L. *IBM J. Res. Dev.* **2000**, *44*, 206–226.

(44) Rupp, M.; Tkatchenko, A.; Müller, K.-R.; Von Lilienfeld, O. A. *Phys. Rev. Lett.* **2012**, *108*, 058301.

(45) Montavon, G.; Rupp, M.; Gobre, V.; Vazquez-Mayagoitia, A.; Hansen, K.; Tkatchenko, A.; Müller, K.-R.; Von Lilienfeld, O. A. *New J. Phys.* **2013**, *15*, 095003.

(46) Hansen, K.; Montavon, G.; Biegler, F.; Fazli, S.; Rupp, M.; Scheffler, M.; Von Lilienfeld, O. A.; Tkatchenko, A.; Muller, K.-R. *J. Chem. Theory. Comput.* **2013**, *9*, 3404–3419.

(47) Hansen, K.; Biegler, F.; Ramakrishnan, R.; Pronobis, W.; Von Lilienfeld, O. A.; Müller, K.-R.; Tkatchenko, A. *J. Phys. Chem. Lett* **2015**, *6*, 2326–2331.

(48) McCulloch, W. S.; Pitts, W. *Bull. Math. Biophys.* **1943**, *5*, 115–133.

(49) Kohonen, T. *Neural Netw.* **1988**, *1*, 3–16.

(50) Abdi, H. *J. Biol. Syst.* **1994**, *2*, 247–281.

(51) Bishop, C. M. *Neural Networks for Pattern Recognition*; Oxford university press, 1995.

(52) Clark, J. W. *Scientific Applications of Neural Nets*; Springer, 1999; pp 1–96.

(53) Ripley, B. D. *Pattern Recognition and Neural Networks*; Cambridge university press, 2007.

(54) Haykin, S. S. *Neural Networks and Learning Machines*; Pearson Upper Saddle River, NJ, USA:, 2009; Vol. 3.

(55) Hinton, G.; Deng, L.; Yu, D.; Dahl, G. E.; Mohamed, A.-r.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T. N. *IEEE Signal Process. Mag.* **2012**, *29*, 82–97.

(56) Simonyan, K.; Zisserman, A. *arXiv preprint arXiv:1409.1556* **2014**,

(57) Lawrence, S.; Giles, C. L.; Tsoi, A. C.; Back, A. D. *IEEE Trans. Neural Netw.* **1997**, *8*, 98–113.

(58) Gybenko, G. *Math. Control Signals Syst.* **1989**, *2*, 303–314.

(59) Hornik, K. *Neural Netw.* **1991**, *4*, 251–257.

(60) Behler, J. *Phys Chem Chem Phys.* **2011**, *13*, 17930–17955.

(61) Manzhos, S.; Carrington Jr, T. *J. Chem. Phys.* **2006**, *125*, 084109.

(62) Manzhos, S.; Carrington Jr, T. *J. Chem. Phys.* **2007**, *127*, 014103.

(63) Malshe, M.; Narulkar, R.; Raff, L.; Hagan, M.; Bukkapatnam, S.; Agrawal, P.; Komanduri, R. *J. Chem. Phys.* **2009**, *130*, 184102.

(64) Handley, C. M.; Popelier, P. L. *J. Phys. Chem. A* **2010**, *114*, 3371–3383.

(65) Yao, K.; Herr, J. E.; Brown, S. N.; Parkhill, J. *J. Phys. Chem. Lett.* **2017**,

(66) Behler, J.; Parrinello, M. *Phys. Rev. Lett.* **2007**, *98*, 146401.

(67) Khorshidi, A.; Peterson, A. A. *Comput. Phys. Commun.* **2016**, *207*, 310–324.

(68) Artrith, N.; Urban, A.; Ceder, G. *Phys. Rev. B* **2017**, *96*, 014112.

(69) Bartók, A. P.; Kondor, R.; Csányi, G. *Phys. Rev. B* **2013**, *87*, 184115.

(70) Rupp, M. *Int. J. Quantum Chem.* **2015**, *115*, 1058–1073.

(71) Behler, J. *Angew. Chem. Int. Ed.* **2017**,

(72) Behler, J. *J. Phys. Condens. Matter* **2014**, *26*, 183001.

(73) Schütt, K. T.; Arbabzadah, F.; Chmiela, S.; Müller, K. R.; Tkatchenko, A. *Nat. Commun.* **2017**, *8*, 13890.

(74) Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A.; Potts, C. Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank. Proc. of the conference on empirical methods in natural language processing (EMNLP). 2013; pp 1631–1642.

(75) Sutskever, I.; Martens, J.; Hinton, G. E. Generating Text with Recurrent Neural Networks. Proc. 28th Annu. Int. Conf. Mach. Learn. (ICML-11). 2011; pp 1017–1024.

(76) Socher, R.; Chen, D.; Manning, C. D.; Ng, A. Reasoning with Neural Tensor Networks for Knowledge Base Completion. Proc. Advances in Neural Information Processing Systems. 2013; pp 926–934.

(77) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; Von Lilienfeld, O. A. *Sci. Data* **2014**, *1*, 140022.

(78) Schütt, K. T.; Kindermans, P.; Sauceda, H.; Chmiela, S.; Tkatchenko, A.; Müller, K. SchNet: A Continuous-Filter Convolutional Neural Network for Modeling Quantum Interactions. Adv. Neural Inf. Process. Syst. 2017.

(79) He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. IEEE Pattern Recogn. 2016; pp 770–778.

(80) In order to avoid possible confusion, it should be noted that throughout ref. 78, the coefficient vectors $\mathbf{c}_i$ are referred to as feature maps $\mathbf{x}_i$ instead. We kept the nomenclature of ref. 73 for clarity and to highlight similarities between the DTNN and SchNet.

(81) Chmiela, S.; Tkatchenko, A.; Sauceda, H. E.; Poltavsky, I.; Schütt, K. T.; Müller, K.-R. *Sci. Adv.* **2017**, *3*, e1603015.

(82) Blanco, M.; Martín Pendás, A.; Francisco, E. *J. Chem. Theory. Comput.* **2005**, *1*, 1096–1109.

(83) Francisco, E.; Martín Pendás, A.; Blanco, M. *J. Chem. Theory. Comput.* **2006**, *2*, 90–102.

(84) Mitoraj, M. P.; Michalak, A.; Ziegler, T. *J. Chem. Theory. Comput.* **2009**, *5*, 962–975.

(85) Harris, D.; Harris, S. *Digital Design and Computer Architecture*; Morgan Kaufmann, 2010.

(86) Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. Adv. Neural Inf. Process. Syst. 2013; pp 3111–3119.

(87) Globerson, A.; Chechik, G.; Pereira, F.; Tishby, N. *J. Mach. Learn. Res.* **2007**, *8*, 2265–2295.

(88) For a definition of the arctan2 function, see Eq. S1.

(89) Neidinger, R. D. *SIAM Rev. Soc. Ind. Appl. Math.* **2010**, *52*, 545–563.

(90) Hahnloser, R. H.; Sarpeshkar, R.; Mahowald, M. A.; Douglas, R. J.; Seung, H. S. *Nature* **2000**, *405*, 947–951.

(91) Dugas, C.; Bengio, Y.; Bélisle, F.; Nadeau, C.; Garcia, R. Incorporating Second-Order Functional Knowledge for Better Option Pricing. Adv. Neural Inf. Process. Syst. 2001; pp 472–478.

(92) Orr, G. B.; Müller, K.-R. *Neural Networks: Tricks of the trade*; Springer, 2003.

(93) Broomhead, D. S.; Lowe, D. *Radial Basis Functions, Multi-Variable Functional Interpolation and Adaptive Networks*; 1988.

(94) Lowe, D. *Complex Syst. 2*, 321–355.

(95) Schwenker, F.; Kestler, H. A.; Palm, G. *Neural Netw.* **2001**, *14*, 439–458.

(96) Klambauer, G.; Unterthiner, T.; Mayr, A.; Hochreiter, S. *arXiv preprint arXiv:1706.02515* **2017**,

(97) Glorot, X.; Bengio, Y. Understanding the Difficulty of Training Deep Feedforward Neural Networks. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. 2010; pp 249–256.

(98) Ruddigkeit, L.; Van Deursen, R.; Blum, L. C.; Reymond, J.-L. *J. Chem. Inf. and Mod.* **2012**, *52*, 2864–2875.

(99) Perdew, J. P.; Burke, K.; Ernzerhof, M. *Phys. Rev. Lett.* **1996**, *77*, 3865.

(100) Tkatchenko, A.; Scheffler, M. *Phys. Rev. Lett.* **2009**, *102*, 073005.

(101) Brooks, B. R. et al. *J. Comput. Chem.* **2009**, *30*, 1545–1614.

(102) Yang, Y.; Meuwly, M. *J. Chem. Phys.* **2010**, *133*, 064503.

(103) Frisch, M. J. et al. Gaussian-09 Revision A.02. Gaussian Inc. Wallingford CT 2009.

(104) Kingma, D.; Ba, J. *arXiv preprint arXiv:1412.6980* **2014**,

(105) Zill, D.; Wright, W. S.; Cullen, M. R. *Advanced Engineering Mathematics*; Jones & Bartlett Learning, 2011.

(106) Abadi, M. et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015; `http://tensorflow.org/`, Software available from tensorflow.org.

(107) Sheather, S. J.; Jones, M. C. *J. Royal Stat. Soc.* **1991**, 683–690.

(108) McNaught, A. D.; McNaught, A. D. *Compendium of Chemical Terminology*; Blackwell Science Oxford, 1997; Vol. 1669.

(109) Anderson, E.; Veith, G. D.; Weininger, D. *SMILES, a Line Notation and Computerized Interpreter for Chemical Structures*; US Environmental Protection Agency, Environmental Research Laboratory, 1987.

(110) Even, S. *Graph Algorithms*; Cambridge University Press, 2011.

(111) Proctor, W.; Yu, F. *Phys. Rev.* **1950**, *77*, 717.

(112) Stock, L. M. *J. Chem. Educ* **1972**, *49*, 400.

(113) Wehril, F.; Marchand, A. P.; Wehrli, S. *Interpretation of Carbon-13 NMR Spectra*; John Wiley & Sons, Inc., 1988.

(114) Lange, N. A. Lange's Handbook of Chemistry. 1967.

(115) Brockherde, F.; Li, L.; Burke, K.; Müller, K.-R. *Nat. Commun.* **2017**, *8*.

(116) Artrith, N.; Morawietz, T.; Behler, J. *Phys. Rev. B* **2011**, *83*, 153101.

(117) Hirshfeld, F. L. *Theor. Chem. Acc.* **1977**, *44*, 129–138.

(118) Piquemal, J.-P.; Gresh, N.; Giessner-Prettre, C. *J. Phys. Chem. A* **2003**, *107*, 10353–10359.

(119) Kramer, C.; Gedeck, P.; Meuwly, M. *J. Comput. Chem.* **2012**, *33*, 1673–1688.

(120) Gastegger, M.; Behler, J.; Marquetand, P. *arXiv preprint arXiv:1705.05907* **2017**,

(121) Morawietz, T.; Behler, J. *J. Phys. Chem. A* **2013**, *117*, 7356–7366.

(122) Grimme, S.; Antony, J.; Ehrlich, S.; Krieg, H. *J. Chem. Phys.* **2010**, *132*, 154104.

(123) Morawietz, T.; Singraber, A.; Dellago, C.; Behler, J. *Proc. Natl. Acad. Sci. USA* **2016**, 201602375.