Technische Universität München
Fakultät für Chemie

Transferability in Chemical Machine Learning

Sina Stocker

Vollständiger Abdruck der von der Fakultät für Chemie der Technischen Universität München zur Erlangung des akademischen Grades einer

Doktorin der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Prof. Dr. Steffen J. Glaser

Prüfer der Dissertation:

1. Prof. Dr. Karsten Reuter
2. Prof. Dr. Harald Oberhofer

Die Dissertation wurde am 30.06.2022 bei der Technischen Universität München eingereicht und durch die Fakultät für Chemie am 04.08.2022 angenommen.

Für Simon.

Preface

This cumulative thesis is based on three papers published in Ref. [1–3]. Two of them have already been published in international peer-reviewed journals. The last one has been submitted to an international peer-reviewed journal. It will not be accepted until submission of this thesis, but is already publicly available as a preprint on ChemRxiv. This thesis aims to give a broader insight and embeds the used papers in an overall context. Furthermore, an introduction to the methods used and the corresponding literature is given. Summaries and author contributions of the papers are also included in this thesis. The papers themselves, as well as supplementary information are attached as appendix. The supporting information to the paper, which is uploaded on ChemRxiv, is also included, but has not been uploaded yet to the platform.

The entire work was carried out between June 2018 and August 2020 at the Chair of Theoretical Chemistry at TU Munich and from September 2020 to June 2022 at the Fritz Haber Institute of the Max Plank Society in Berlin, under the supervision of Prof. Dr. Karsten Reuter. A research stay at the Engineering Department at the University of Cambridge in October 2019 hosted by Prof. Dr. Gábor Csányi completes this work.

Sina Stocker, Berlin, June 2022

Abstract

The combination of machine learning (ML) and computational chemistry offers unprecedented opportunities to gain new insights into chemical processes. Established computational chemistry methods are often either too computationally demanding or do not provide the required accuracy. Machine learning methods might overcome these limitations and are able to predict physical properties very accurately, but significantly cheaper than quantum mechanical (QM) methods. However, the generation of large reference databases, which are often required for training ML models, is still a computationally costly task. This results in the fact that only few of such large databases exist that cover certain sub-parts of the chemical space. The focus of this thesis is therefore to explore the transferability of ML models trained on such fixed databases, but by applying them to predictions on other subsets of chemical or reaction space. This exploration will be shown and discussed on the basis of three different examples.

In the first example, established ML methods in chemical compound space were used to predict reaction energies in chemical reaction space. The predicted reaction energies can then be utilized to explore and reduce complex reaction networks. As a first step, a QM-based reference database of closed-shell molecules and radical systems has been generated to describe chemical reactions. Moreover, the analysis demonstrated that for adequate predictions in reaction space, certain requirements have to be satisfied for compound space ML methods to ensure transferable models. The resulting model could be used for the non-empirical reduction of reaction networks, with methane combustion as an example.

The second example focused on the exploration of different parts of the chemical space with molecules of large size differences. An important requirement for this is the use of size-extensive ML models. To this end, this part of the thesis showed how size-extensive ML models can be build to satisfactorily predict properties of large molecules, when training on small systems. The results further showed, that non size-extensive models completely failed in that task.

In the last example, the robustness of advanced graph neural network (GNN) models in atomistic simulations was investigated. To this end, models were trained on different subsets of the fixed QM7-x database. This is an interesting test scenario, as the capabilities of GNNs were mostly tested on established databases, whereas fewer studies have been conducted to show their applicability in chemical simulations. The results showed that stable dynamics could be achieved for GNN models trained on large training set sizes. Furthermore, it was found that instabilities during the simulations could occur, even though the model produces low errors on a fixed test set.

Zusammenfassung

Die Kombination aus maschinellem Lernen (ML) and computergestützter Chemie bietet zuvor noch nie dagewesene Möglichkeiten ein tieferes Verständnis über chemische Vorgänge zu erlangen, da etablierte Methoden der theoretischen Chemie oft entweder zu rechenintensiv sind oder nicht die geforderte Genauigkeit liefern. Maschinell gelernte Modelle können physikalische Eigenschaften sehr genau vorhersagen, wobei sie eine kostengünstige Alternative zu quantenmechanischen (QM) Methoden darstellen. Die Erzeugung großer Referenzdatenbanken, die für das Trainieren von ML Modellen benötigt werden, stellt sich jedoch aus Kostengründen noch als schwierig heraus, was zur Folge hat, dass nur wenige solcher großen Datenbanken existieren. Diese decken dabei jeweils auch nur einen gewissen Teilbereich des "Chemical Space" ab. Der Fokus dieser Thesis liegt daher auf der Untersuchung der Transferierbarkeit von ML Modellen, die auf solch großen Datenbanken trainiert wurden. Die Vorhersagen werden jedoch für andere Teile des "Chemical Space" oder "Reaction Space" getroffen. Dies wird anhand dreier unterschiedlicher Beispiele gezeigt und diskutiert.

Im ersten Beispiel wurden etablierte ML Methoden des "Compound Space" verwendet, um Reaktionsenergien im "Reaction Space" vorherzusagen. Die vorhergesagten Reaktionsenergien können dann genutzt werden, um komplexe Reaktionsnetzwerke zu untersuchen oder um sie zu vereinfachen. Für die Vorhersage von Reaktionsenergien wurde im ersten Schritt eine QM-basierte Referenzdatenbank von geschlossenschaligen Molekülen und Radikalen erstellt. Diese Datenbank ist nötig, um chemische Reaktionen beschreiben können. Die Analysen zeigten, dass für angemessene Vorhersagen im "Reaction Space" bestimmte Anpassungen für die ML Methoden im "Compound Space" gemacht werden müssen, um übertragbare Modelle zu gewährleisten. Das finale Modell konnte für die nicht-empirische Reduktion von Reaktionsnetzwerken am Beispiel der Methanverbrennung verwendet werden.

Das zweite Beispiel setzte den Fokus auf die Erkundung verschiedener Teilbereiche des "Chemical Space" für Moleküle mit erheblichen Größenunterschieden. Eine wichtige Voraussetzung für diese extrapolierende Erkundung ist dabei die Verwendung von extensiven ML Modellen. Die Erzeugung dieser wurde daher in diesem Teil der Arbeit dargestellt und diskutiert. Es wurde gezeigt, dass extensive ML Modelle zufriedenstellende Vorhersagen für große Moleküle liefern können, wenn die Modelle zuvor auf kleine Systeme trainiert wurden. Die Ergebnisse zeigten außerdem, dass nicht extensive Modelle bei dieser Aufgabe völlig versagen.

Im letzten Beispiel wurde die Robustheit von sogenannten "Graph Neural Networks" (GNNs) in atomistischen Simulationen untersucht. Diese neuronalen Netze wurden zuvor auf verschiedenen Teilmengen der QM7-x Datenbank trainiert. Die Untersuchungen sind in sofern interessant, da die Performance von GNNs meist auf etablierte Datenbanken getestet wird. Es wurde bislang jedoch noch weniger ihre Anwendbarkeit in chemischen Simulationen untersucht. Die Ergebnisse zeigten, dass stabile Simulationen mit GNNs erreicht werden können, wenn diese Modelle mit großen Datenmengen trainiert werden. Darüberhinaus wurde festgestellt, dass Instabilitäten während den Simulationen auftreten können, auch wenn die Modelle kleine Fehler auf sogenannten Testdatensätzen liefern.

Contents

List of Abbreviations

Δ-ML	Delta machine learning
AE	Atomization energy
AI	Artificial intelligence
AIMD	*Ab initio* molecular dynamics
ANN	Artificial neural network
BOA	Born Oppenheimer approximation
DFT	Density functional theory
FF	Force field
FPS	Farthest-point sampling
GGA	Generalized gradient approximation
GNN	Graph neural network
GPR	Gaussian process regression
iMBTR	Intensive many-body tensor representation
KRR	Kernel ridge regression
KS-DFT	Kohn-Sham density functional theory
LDA	Localized density approximation
MBTR	Many-body tensor representation
ML	Machine learning
MLIP	Machine learning interatomic potential
MP	Message passing
MPNN	Massage passing neural network
NN	Neural network
PBE	Perdew–Burke-Ernzerhof
PES	Potential energy surface
QM	Quantum mechanics

RBF Radial basis function
RE Reaction energy

SCF Self-consistent field
SE Schrödinger equation
SOAP Smooth overlap of atomic positions

TS Transition state

UFF Universal force field

1 Introduction

Chemists are interested in the composition, properties and reactions of chemical compounds. In this respect, a microscopic picture on the atomistic level provides fundamental insights into a chemical structure or process and allows for the discovery of new materials with favourable properties [4–7] and great value for both technology and industry. The understanding of individual molecular reaction paths in chemical processes, for example, may help to find better catalysts that facilitate the production of industrially relevant products or for the development of renewable fuels that contribute to a sustainable future. [8–11] In this context, computer-aided calculations and simulations are essential tools to provide specifically these insights into the chemical structures and processes as the search space for finding new compounds can be incredible vast [12, 13].

In many respects, first principles quantum mechanical (QM) methods form the basis for atomistic studies in computational chemistry. Such QM calculations are typically based on the numerical solution of the Schrödinger equation formulated within the Born-Oppenheimer approximation [14] and allow for the determination of accurate ground-state properties for given geometries. Unfortunately, the computational cost for evaluating physical properties with electronic structure methods, such as density functional theory (DFT) [15–17], quickly becomes prohibitively expensive and therefore limits the calculations to relativity small system sizes. This expense factor is particularly impractical in high throughput screenings. Furthermore, only ps time scales are feasible, when propagating the chemical system over time in so-called *ab initio* molecular dynamics simulations [18]. These limitations pose significant challenges for the simulation of realistic chemical systems and processes.

To circumvent these limitations, empirical force fields (FF) with simple analytic expressions are often used and have been parametrized to simulate the system of interest at desired length and time scales [19–21]. Their usage, however, involves some disadvantages regarding predictive accuracy and parameterization. On the one hand, empirical FF are less accurate in contrast to DFT, whereby this lack in accuracy can be problematic to quantitatively model molecular interactions or chemical reactions. On the other hand, finding appropriate functional forms and parametrizations usually results in a difficult and tedious quest [22–25]. A solution to both disadvantages is offered by modern machine learning (ML) methods, which have been proven to perform highly accurate predictions—when trained on QM data—without deciding for a predefined functional form. [25–27]

An ML algorithm can in principle learn any complex relationship between input and output values from a training database. Computational chemists have taken advantage of this ability and use ML methods as surrogate models to fit structure-property relationships. A commonly applied example for this is the expression of the potential energy (and/or forces) as a function of atomic positions, which is in this context denoted as machine learning interatomic potentials (MLIPs) [28–30]. Learning such relations typically requires an encoding of the molecular structures from a reference database and subsequently learning the structure-energy relationship with kernel [31–35] or neural network (NN) [36–42] based regression models. These methods are able to simulate chemical systems at desired time and length scales on QM accuracy, however, with a fraction of the cost of the underlying reference method. [43–46] Besides structure-energy

relationships, other properties such as dipole moments [47], band gaps[48], polarizabilities [49, 50], enthalpies [51] or binding energies on surfaces [52] have been successfully predicted by ML methods and consequently demonstrated their applicability for computational chemistry questions also in a high throughput fashion.

The training database plays an essential role for the successful generation of an ML model and its quality of the predictions. In case this database does not contain essential features and representative configurations relevant for the predictions, it is not assumed to lead to an adequate ML model. In contrast to applications such as image recognition [53] or natural language processing [54], where big data is available, this is typically not the case in computational chemistry. Quantum mechanical reference calculations are computationally expensive and the dimensionality of chemical compound or reaction spaces including many elements across the periodic table is tremendously large.

Nonetheless, QM-based databases for certain sub-parts of the chemical space exist, which are predominantly generated for molecular structures. Prominent examples for such databases are the QM9 [55, 56] or the QM7-x [57] data set consisting of around 134,000 and 4.2 million configurations of small closed-shell organic molecules, respectively. While the QM9 data set provides only equilibrium structures of molecules consisting of up to nine non-hydrogen atoms (C, O, N, F), QM7-x contains both equilibrium and non-equilibrium configurations with molecules up to seven non-hydrogen atoms (C, O, N, S, Cl). Both databases provide various DFT calculated properties such as energies and forces for each configuration. However, even these databases with more than 100,000 or millions of structures are rather sparse approximations of the chemical compound space, considering that the chemical compound space of drug relevant molecules is estimated to contain more than 10^{60} molecules [58].

Nevertheless, if a fixed database is available or has been generated that covers important sub-parts of the chemical space, an intriguing next step would be to examine the corresponding ML models in the extrapolative regime, i.e. when applying them to other sub-parts of chemical or even reaction space. In this context, ML models are trained on fixed databases and extrapolated predictions can be made without—or at least with only a few—additional expensive reference calculations, which leads to data-efficient approaches. This cumulative thesis addresses this scientific issue and explores the transferability of chemical machine learning methods in the following three (published) examples. An illustrative overview about these examples is given in Fig. 1.1.

In our first example published in [1], we transfer established atomization energy (AE) predictions in chemical compound space to the calculations of reaction energies (RE) in reaction space for the exploration and rational reduction of large and complex reaction networks. Our second publication [2] addresses size-extensivity issues in ML models with so-called global representations. Here, we were particularly interested in the exploration of ML methods trained on small molecules and tested on significantly larger configurations to explicitly show the importance of size-extensivity for data-efficient models. Finally, in the third contribution [3], we explore the applicability of modern graph neural network (GNN) MLIPs [59] in long and hot molecular dynamics (MD) simulations, when trained on QM7-x. This rather novel class of MLIP produces highly accurate predictions on benchmark data sets, which is a commonly applied test scenario for GNNs. Their robustness in dynamic studies has been, however, less explored.

To put the work contained in this cumulative thesis into a broader perspective, the subsequent chapters will present the theoretical background for the here used methods to provide a comprehensive picture of ML methods and their application in computational chemistry. Therefore,

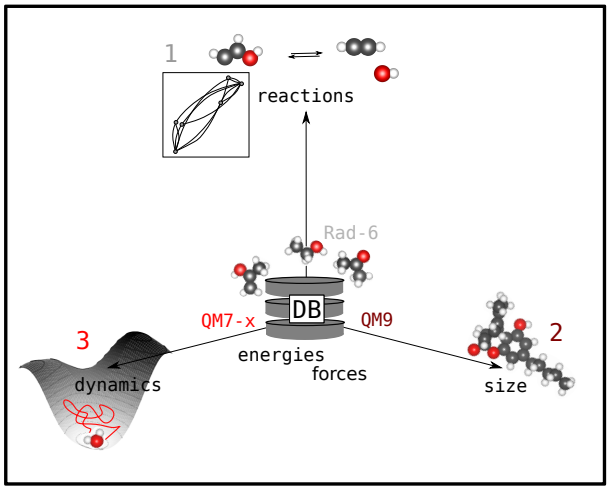


Fig. 1.1: Cumulative Thesis Overview. *The figure provides a pictorial overview of three examples for testing the transferability of machine learning models examined in this thesis. The focus has been set on ML models that have been trained on benchmark databases (Rad-6, QM9, QM7-x) and the performance was explored and tested in extrapolated regimes. This includes atomization energy (AE) predictions on the Rad-6 database and corresponding calculations of reaction energies (RE) for given reaction networks in the first example (1). In the second example, we use rather 'small' molecules from the QM9 database in the training procedure and evaluate the prediction error on significantly larger molecules to address size-extensivity issues in the use of global representations (2). Finally, we use the QM7-x database to train a graph neural network (GNN) and explore the corresponding robustness of the potential in long and hot MD simulations (3) in the third example.*

chapter 2 will introduce the reader to the fundamental concepts of theoretical chemistry, starting with the quantum mechanical treatment of molecular systems and the resulting concept of the potential energy surface (PES). This is followed by the explanation of DFT required to numerically calculate the PES. Furthermore, the basic concepts of MD simulations are introduced, which is a method to dynamically explore the PES.

In this thesis, a particular focus has been set on ML models and the generation of MLIP in computational chemistry. Therefore, an extensive overview about ML is given in chapter 3, which includes the generation of a training database, the representations of molecular geometries, the learning algorithms for both NN and kernel-based ML and the appropriate validation of respective ML models.

Afterwards, content summaries and the assignment of individual author contributions of the relevant publications are given in chapter 4. The thesis is completed with a conclusions and outlook chapter. All thesis relevant publications are attached as appendix.

2 Theoretical Basics of Computational Chemistry

As already introduced, this thesis focuses on the transferability of ML methods in computational chemistry applications. Therefore, this chapter provides the reader with the fundamental chemical concepts required to build these ML models. In particular, the focus is on an introduction to those theoretical chemistry methods applied during the work of this thesis. In principle, this can be summarized in terms of the following two aspects: On the one hand, the reader will be introduced into quantum mechanical computations to build accurate reference databases, from which the ML algorithm learns. And on the other hand, a short introduction into atomistic simulations is given, from which physical observables can be derived using ML methods.

2.1 Potential Energy Surface

Computational chemists use atomistic simulations to gain fundamental insights into the constitution and properties of matter or to study mechanistic processes of chemical reactions. In case of the latter, an illustrative example is given by a simple dissociation reaction where one molecule cleaves into two smaller fragments through a bond breaking event. Studying such an atomistic process, in which nuclei and electrons are involved, requires a quantum mechanical treatment. The central equation in quantum mechanics and therefore computational chemistry, which models the interactions between N nuclei and n electrons in a chemical system, e.g. a molecule, is the (non-relativistic) time-independent Schrödinger equation (SE)

$$\hat{H}\,\Psi(\boldsymbol{r},\boldsymbol{R}) = E\,\Psi(\boldsymbol{r},\boldsymbol{R})\,. \tag{2.1}$$

Here, $\hat{H}$ is the Hamiltonian, E is the energy of the chemical system and $\Psi(\boldsymbol{r},\boldsymbol{R})$ is the wavefunction. The latter depends on both: the nuclear ($\boldsymbol{R}$) and electronic ($\boldsymbol{r}$) coordinates. To be more specific, the Hamiltonian is defined as a sum of kinetic $\hat{T}$ and potential energy $\hat{V}$ operators, which is

$$\hat{H} = \hat{T}_{\text{el}} + \hat{T}_{\text{Nucl}} + \hat{V}_{\text{el-el}} + \hat{V}_{\text{el-Nucl}} + \hat{V}_{\text{Nucl-Nucl}}, \tag{2.2}$$

where the subscript el and Nucl describe the electronic and nuclear contributions, respectively.

Although the SE provides a recipe to calculate the systems energy, it is only analytically solvable for very simple systems, like the hydrogen atom. As a consequence, for more complex systems, such as molecules, approximations are required that simplify the SE. In this context, the adiabatic Born-Oppenheimer approximation (BOA) [14] is often applied, postulating the separation of electronic and nuclear variables. Since electrons are much lighter in mass compared to nuclei, they move faster. Thus, they are able to immediately follow the motions of the nuclei. This assumption enables the formulation of an electronic Hamiltonian

$$\hat{H}_{\text{el}} = \hat{T}_{\text{el}} + \hat{V}_{\text{el-el}} + \hat{V}_{\text{el-Nucl}}\,, \tag{2.3}$$

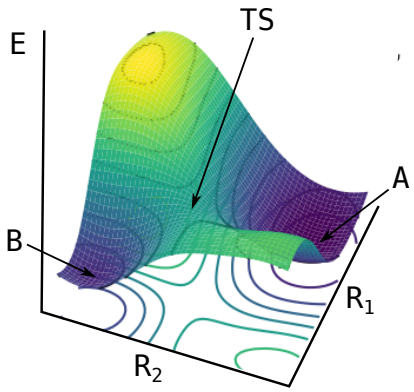


Fig. 2.1: Potential Energy Surface. *Illustrative example of a two-dimensional PES. Chemical relevant points are marked with arrows accordingly. A and B correspond to minimum configurations. The saddle point represents the transition state (TS) separating both equilibrium geometries via the lowest energy path.*

which depends only on the electronic variables. Nuclear contributions are added parametrically. Solving the resulting electronic SE

$$\hat{H}_{\text{el}}\Psi(\boldsymbol{r}) = E_{\text{el}}\,\Psi(\boldsymbol{r}) \tag{2.4}$$

leads to the electronic ground state energy E_{el} of the system and defines the potential energy surface (PES), which builds the basis for most atomistic simulations. [60–63]

The PES is a $3N$ dimensional hypersurface, where N is the number of atoms in the system. Since the energy is invariant to global translations and rotations, this typically reduces to $3N$-6 dimensions. Of particular interest are certain points on the PES that correspond to specific configurations of the chemical system. Local minima for example, correspond to meta-stable states. Gradients of the potential energy with respect to atomic coordinates (i.e. forces) can be used to perform local geometry optimizations, ending up in a nearby minimum configuration (equilibrium configuration). Other important points are saddle points that belong to transition state (TS) geometries. They connect minimum configurations via lowest energy paths in chemical reactions. The energy barrier that has to be overcome in such a reaction is denoted as the reaction barrier. It corresponds to the energy—relative to the reactant—that is required to form the TS. This reaction barrier as well as the temperature, at which a reaction takes place determine the speed of a chemical reaction and thus the reaction rate that allows for studying reaction kinetics. Knowing the entire PES of a chemical system enables the exploration of stable configurations and corresponding reactions among them. Figure 2.1 gives an illustrative example of a fictional PES, showing respective chemical meaningful points. [60, 63]

2.2 Density Functional Theory

In this section, we address the question of how the electronic SE (Eq. 2.4) can be solved in a practical way and which machinery can be applied to obtain the energy of a molecular system. A commonly applied approach in computational chemistry is density functional theory (DFT). This method is based on the three-dimensional electronic density $\rho(\boldsymbol{r})$ instead of the high dimensional wavefunction. Thereby, the systems energy is expressed as an energy functional in terms of the density $E[\rho(\boldsymbol{r})]$. The main concepts of DFT are introduced in the following.

Fundamentals in DFT are given by the two Hohnberg-Kohn [16] theorems. The first theorem defines a relation between the electronic ground state density $\rho_0(\boldsymbol{r})$ and the ground state energy of the system. In addition, the second Hohnberg-Kohn theorem shows that there is a unique energy functional $E\left[\rho_0(\boldsymbol{r})\right]$, which leads to the ground state energy of the system. Specifically, this means that any trial density $\rho(\boldsymbol{r})$, non equal to $\rho_0(\boldsymbol{r})$, leads to a higher energy of the chemical system. The second theorem is mathematically expressed as:

$$E_0 = E\left[\rho_0(\boldsymbol{r})\right] < E\left[\rho(\boldsymbol{r})\right] \text{ with } E\left[\rho(\boldsymbol{r})\right] = T_{\text{el}}\left[\rho(\boldsymbol{r})\right] + V_{\text{el-Nucl}}\left[\rho(\boldsymbol{r})\right] + V_{\text{el-el}}\left[\rho(\boldsymbol{r})\right], \quad (2.5)$$

where $T_{\text{el}}\left[\rho(\boldsymbol{r})\right]$ represents the kinetic energy functional and $V_{\text{el-Nucl}}\left[\rho(\boldsymbol{r})\right]$ and $V_{\text{el-el}}\left[\rho(\boldsymbol{r})\right]$ the potential energy functionals for the electron-nuclei and electron-electron interactions, respectively. Equation 2.5 provides an expression to obtain the system's energy in terms of the electronic ground state density. The crux, however, is that only the potential energy term $V_{\text{el-Nucl}}\left[\rho(\boldsymbol{r})\right]$ is mathematically accessible. Formulations and approximations for $T_{\text{el}}\left[\rho(\boldsymbol{r})\right]$ and $V_{\text{el-el}}\left[\rho(\boldsymbol{r})\right]$ are derived in the following. [15]

The basics have been established by Kohn and Sham in so-called Kohn-Sham DFT [17] (KS-DFT). There, the first fundamental assumption is to express large parts of $T_{\text{el}}\left[\rho(\boldsymbol{r})\right]$ and $V_{\text{el-el}}\left[\rho(\boldsymbol{r})\right]$, for which formulations in terms of the density are mathematically accessible. Missing contributions are stored in an additional functional: the exchange correlation functional $E_{\text{XC}}[\rho(\boldsymbol{r})]$. By this assumption Eq. 2.5 can be reformulated as:

$$\begin{aligned} E[\rho(\boldsymbol{r})] &= V_{\text{el-Nucl}}\left[\rho(\boldsymbol{r})\right] + \underbrace{T_{\text{N}}[\rho(\boldsymbol{r})]}_{\text{large}} + \underbrace{J[\rho(\boldsymbol{r})]}_{\text{large}} + E_{\text{XC}}[\rho(\boldsymbol{r})] \text{ with} \\ E_{\text{XC}}[\rho(\boldsymbol{r})] &= \underbrace{\left(T_{\text{el}}\left[\rho(\boldsymbol{r})\right] - T_{\text{N}}\left[\rho(\boldsymbol{r})\right]\right)}_{\text{small}} + \underbrace{\left(V_{\text{el-el}}\left[\rho(\boldsymbol{r})\right] - J\left[\rho(\boldsymbol{r})\right]\right)}_{\text{small}}. \end{aligned} \quad (2.6)$$

Here, $J[\rho(\boldsymbol{r})]$ is the classical Coulomb energy in terms of the density for the electronic interactions

$$J[\rho(\boldsymbol{r})] = \frac{1}{2}\int \frac{\rho(\boldsymbol{r})\rho(\boldsymbol{r'})}{|\boldsymbol{r}-\boldsymbol{r'}|} d\boldsymbol{r}d\boldsymbol{r'}. \quad (2.7)$$

Everything that is not covered by the classical Coulomb energy, i.e. non-classical contributions, such as electron correlation and exchange effects or self-interaction corrections, go into the exchange correlation functional.

To express parts of the kinetic energy functional, a fictitious reference system of non-interacting electrons was introduced. The special feature of this auxiliary system is that its density is constructed in a way to be in accordance with the true ground state density of the system with interacting electrons. By introducing this reference system, the kinetic energy functional $T_N[\rho(\boldsymbol{r})]$ of the non-interacting system can be calculated. Again missing parts of kinetic energy contributions are stored in $E_{XC}[\rho(\boldsymbol{r})]$.

In KS-DFT, the electrons are defined by single-electron orbitals $\psi_i(\boldsymbol{r})$, from which a Slater determinant is constructed, representing the ground state of the non-interacting system. These orbitals are used to calculate the density of the non-interacting system (ρ_{KS}) and thus the true ground state density of the real system

$$\rho_0(\boldsymbol{r}) = \rho_{KS}(\boldsymbol{r}) = \sum_i^n |\psi_i(\boldsymbol{r})|^2 . \tag{2.8}$$

As a consequence of the non-interacting system, it can be written as a sum of n single-particle Hamiltonian operators $\hat{h}_{KS,i}$

$$\hat{h}_{KS,i} = -\frac{1}{2}\nabla_i^2 + v_{eff}(\boldsymbol{r}), \text{with } v_{eff}(\boldsymbol{r}) = \sum_a^N \frac{Z_a}{|\boldsymbol{r} - \boldsymbol{R}_a|} + \int \frac{\rho(\boldsymbol{r'})}{|\boldsymbol{r} - \boldsymbol{r'}|} d\boldsymbol{r'} + V_{xc}. \tag{2.9}$$

The first term in the Hamiltonian defines the kinetic energy of the auxiliary system. The second term denotes an effective potential that is constructed in a way to ensure that both densities (interacting and non-interacting densities) are equal. This construction is mathematically defined as a sum of two Coulomb interaction terms and an exchange correlation potential. While the first Coulomb term represents the interactions between an electron and all N nuclei with proton number Z_a, the second Coulomb term describes the interactions between electrons. The latter is formulated as the interaction between an electron and a mean field electronic density $\rho(\boldsymbol{r'})$ created by all electrons in the system. Lastly, the exchange correlation potential V_{xc} is defined as the derivative of the exchange correlation energy $\frac{\partial E_{xc}[\rho(\boldsymbol{r})]}{\partial \rho(\boldsymbol{r})}$.

The resulting n Kohn-Sham equations (coupled via the electron density)

$$\left(-\frac{1}{2}\nabla_i^2 + v_{eff}(\boldsymbol{r})\right)\psi_i(\boldsymbol{r}) = \epsilon_i \psi_i(\boldsymbol{r}) \tag{2.10}$$

can be transformed into a matrix form. They are solved by applying the second Hohenberg-Kohn theorem in combination with a self-consistent field (SCF) approach. Within the SCF, the systems energy is minimized iteratively by varying the density until a convergence criterion is reached. Kohn-Sham DFT formally scales with $\mathcal{O}(n^3)$. This, however, may differ in various types of implementations. [15, 26, 61–63]

So far, functional expressions in terms of the density have been formulated for all energy terms with the exception of one: the $E_{XC}[\rho(\boldsymbol{r})]$ functional. As seen before, this term contains the unknown parts of the kinetic energy and non-classical energy contributions. In case an expression for $E_{XC}[\rho(\boldsymbol{r})]$ is mathematically accessible, Eq. 2.10 could be solved exactly. So far, it has not been possible to formulate such an exact expression. Consequently, approximate forms are required and different functionals have been developed varying in computational cost and accuracy to be sufficient for different types of chemical applications.

Examples for exchange-correlation functionals are the localized density approximation (LDA) or the generalized gradient approximation (GGA) functionals. These families of LDA and GGA functionals include local density information (the former) as well as information about the gradient of the density (the latter). However, they do not go beyond, i.e. they do not include non-local information.

Another class of functionals are so-called hybrid functionals, which express exchange-correlation terms by mixing parts of the exact Hartree-Fock (HF) exchange energy (for more details about HF see Ref. [60, 62, 63]) into LDA or GGA functional expressions. Mixing exact exchange energy from HF into these functionals reduces parts of the self-interaction error present in the exchange-correlation functional due to the separation of $V_{\text{el}-\text{el}}$ into a classical and a non-classical term. An example hybrid functional that is often applied for molecular systems and also during the work of this thesis in [1, 3] is the PBE0 functional [64, 65]. This functional mixes parts of the Perdew–Burke-Ernzerhof [66] (PBE, GGA) exchange energy with the HF exchange energy

$$E_{\text{XC}}^{\text{PBE0}} = 0.25E_{\text{X}}^{\text{HF}} + 0.75E_{\text{X}}^{\text{PBE}} + E_{\text{C}}^{\text{PBE}}. \tag{2.11}$$

This leads to an improvement of describing many molecular properties in contrast to the PBE functional and is therefore a suitable choice for the generation of molecular reference databases. [15, 26, 60, 63]

2.3 Molecular Dynamics Simulations

So far, fundamentals in computational chemistry have been introduced, i.e. the concept of the PES and how it can be obtained by approximately solving the electronic SE. Density functional theory provides potential energy calculations of chemical systems with fixed nuclear positions. These calculations yield an energy description at zero temperature. However, computational chemists are often interested in atomistic processes and want to mimic them *in silico* to gain insights into these systems. Such processes, like catalytic reactions on a surface or molecule/enzyme interactions take place at finite temperatures. Therefore, simulation techniques are needed allowing to follow the dynamical motions of a chemical system or to calculate ensemble averaged properties (e.g. free energies) at finite temperatures and pressures.

The prototypical way of propagating a chemical system over time is to use a molecular dynamics trajectory (MD), in which molecular positions are adjusted by following Newtons equations of motion

$$\boldsymbol{F}_a(t) = -\left(\frac{\partial V(\boldsymbol{R})}{\partial \boldsymbol{R}_a}\right) = m_a \ddot{\boldsymbol{R}}_a. \tag{2.12}$$

Here $\boldsymbol{F}$ is the force that acts on atom a, V is the potential energy, m is the mass and $\boldsymbol{R}$ are the atomic positions in Cartesian coordinates. $\ddot{\boldsymbol{R}}$ denotes the second derivative of the atomic positions with respect to time t. [61] Equation 2.12 describes a classical evolution of the atomic positions. This description is valid for many chemical applications, as quantum nuclear effects can often be neglected. Specifically, this is true when dealing with non hydrogen atoms. [67]

Another interesting aspect to discuss is the realization of conserved physical variables, such as the temperature, during the simulation. This aspect is important to deduce experimentally observed thermodynamic properties. These properties are usually obtained as time averages, when the system is simulated in a statistical ensemble. The canonical ensemble (NVT), for example,

mimics experimental conditions where the particle number, the volume and the temperature are conserved during the simulation. A conserved temperature is realized by coupling the simulated system to an external heat bath through a thermostat.

An example thermostat is the Langevin thermostat [68], which was also used in molecular dynamics simulations during the work of this thesis in [3]. Here, temperature conservation is achieved through stochastic collisions between the simulated system and an imaginary heat bath. Langevin dynamics are modeled by adding a friction and a random collision term to Newton's equations of motions:

$$m_a \ddot{\boldsymbol{R}}_a = -\left(\frac{\partial V(\boldsymbol{R})}{\partial \boldsymbol{R}_a}\right) - \underbrace{\gamma_\mathrm{f} m_a \dot{\boldsymbol{R}}_a}_{\text{friction force}} + \underbrace{\boldsymbol{\eta}_\mathrm{R}(t)}_{\text{random force}} \quad . \tag{2.13}$$

Following the systems evolution over time by integrating the respective Langevin equations, leads to configurations corresponding to the canonical ensemble. [69, 70]

In a MD trajectory, atomic forces have to be evaluated at every time step. These forces can be obtained as gradients of the potential energy in terms of atomic positions evaluated with DFT. Consequently, producing such *ab initio* molecular dynamics (AIMD) trajectories could be computationally very demanding, since it requires the approximate solution of the SE at every time step. Thus, system sizes in AIMD simulations are limited to a few hundreds of atoms and only ps simulation times are feasible. As a consequence, other approaches are required to approximate energy and force calculations in MD simulations or other atomistic simulation techniques.

Here, machine learning methods are posed to revolutionize the field of computational chemistry. Machine learning interatomic potentials can predict molecular quantities very accurately when fitted to quantum mechanical references. Moreover, they do not rely on empirical functions, but instead learn structure-property relationships from reference data. As these methods seem to be impressive tools, the next chapter will give a detailed overview about the current state of ML methods in computational chemistry. [25]

3 Machine Learning in Computational Chemistry

Over the past decades machine learning (ML) methods and artificial intelligence (AI) have become very popular in many fields of our daily lives and we experience a veritable "ML-boom" at the moment. Machine learning algorithms are fundamental especially in these days of social media, however, they are also applied in various other fields such as image classification [53], natural language processing [54], robotics [71], energy economics and finance [72]. With an ML algorithm it is possible to recognize similarities, regularities or relevant patterns in a given data set. This knowledge is then used to define a model predicting the properties of unknown samples or clustering data. Being impressed by the success of ML algorithms in many diverse fields, natural scientists started to use ML methods in scientific research hoping that for instance these methods could generate computationally efficient interatomic potentials with the accuracy of quantum mechanical (QM) methods. [26, 30]

To start with a more general introduction, ML can be divided in several types of learning. One way of learning is the so-called **supervised machine learning**. Here, the user is looking for an ML model f that maps a set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{M}$ of given input values (training data) $\boldsymbol{x}_i$ to respective outputs (labels) y_i, where M is the number of observations. In case the output values y_i are featured by a set of classes or categories, the learning task is denoted as a classification problem. A solution to this problem can give several ML algorithms, such as partial least squares-discriminant analysis or support vector machines.

Another well-known representative for supervised ML is regression. Within a regression task the inputs are fitted to a continuous function. Prominent ML algorithms used for regression are ridge regression, kernel ridge regression (KRR), Gaussian process regression (GPR), artificial neural networks (ANNs) or support vector regression. As we will see in the following, regression can be used to learn potential energies and forces as a function of atomic positions for chemical systems.

Unsupervised learning is a further discipline of ML. Here, unlabeled input values $\boldsymbol{x}_i$ enter the machine learning algorithm to find some latent features in the data or to cluster the data. This includes also the reduction of dimensionality for data visualisation or data prepossessing for other ML tasks like classification. Specifically, the former (dimensionality reduction) was also applied in [1] during the work of this thesis for visualization purposes. Unsupervised ML methods are principal component analysis, kernel principal component analysis, k-means clustering and self-organizing maps. Figure 3.1 summarizes the main differences of supervised and unsupervised learning and gives a pictorial overview. [26]

In the interests of completeness, **reinforcement learning** is a third class of learning, using in the training process some kind of reward system. However, reinforcement learning was not used in the work of this thesis, consequently no further details are provided. The interested reader is referred to the literature [73].

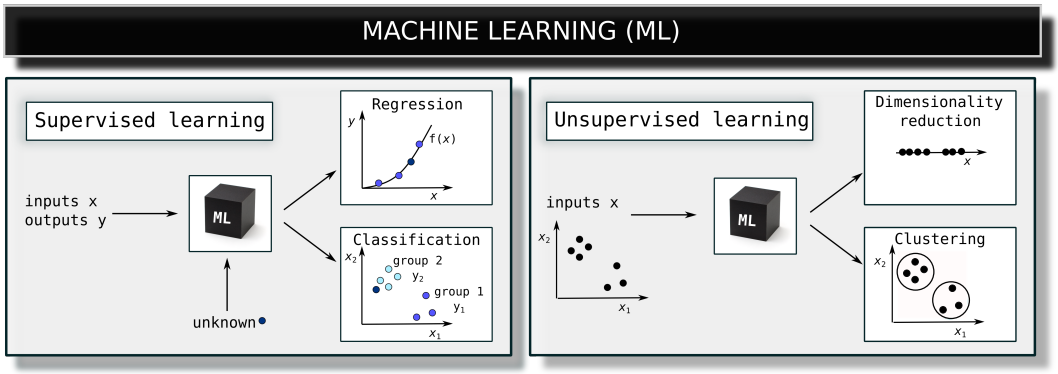


Fig. 3.1: Types of Machine Learning. *The figure summarizes the differences between supervised (left) and unsupervised ML (right). Supervised ML produces models learned from labeled training data ($\boldsymbol{x}_i, y_i$). After training, the ML model is able to predict the function value (regression task) or class label (classification task) of unknown samples. Unlabeled data sets are used in unsupervised ML. Depending on the application, the algorithm either reduces the dimensionality of the database or identifies clusters in the data set.*

In the foregoing paragraphs, the different types of learning have been introduced and we have already mentioned that learning potential energies in terms of atomic coordinates is a regression task. It remains to be clarified, which key ingredients are required to generate machine learning models in computational chemistry. Figure 3.2 gives an overview to that issue. On the one hand, it illustrates the main features needed to learn a relationship between molecular properties and respective molecular configurations. On the other hand, it simultaneously outlines the major topics that will be discussed in this chapter and further have been used during the work of this thesis in [1–3].

For clarification, models aiming to represent a structure-energy (force) relation are denoted as MLIPs in the following, as they fit the PES and thus avoid solving the SE. This applies regardless of whether the model is robust and stable in MD simulations, i.e. it represents the entire PES including forces as gradients of the potential energy [3]. It further applies to models that are specifically able to predict specific sub-spaces of the PES, such as equilibrium configurations [1, 2].

Figure 3.2 illustrates the general workflow required for generating a MLIP. First, we need a representative reference database containing molecular configurations and respective QM properties. Second, a mathematical representation (descriptor) is required to encode the atomic structures and make them accessible as inputs for the ML algorithm. Based on these representations and corresponding labels (e.g. energies and/or forces), the ML model is trained via a regression algorithm (e.g. with KRR or ANNs) in a third step. Note, that graph neural networks (GNNs) are able to simultaneously learn atomic representations and perform the regression. Then, the ML model is validated to ensure appropriate performance for the respective task. Finally, the ML model is applied in the respective task. [26, 31]

Machine learning is more and more used in theoretical chemistry and a lot of progress has already been made. Specifically, this is reflected in excellent review papers [25–27, 74] that arose during the last two years. The current chapter is mainly based on these sources.

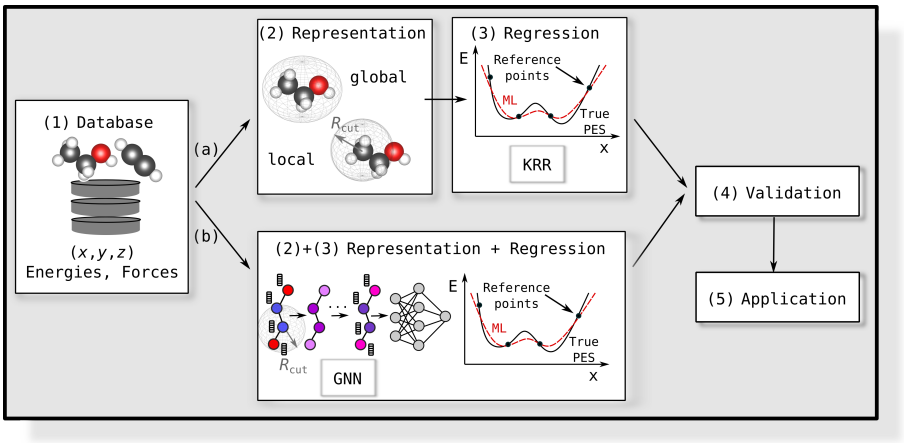


Fig. 3.2: Overview Machine Learning Interatomic Potentials. *The figure gives an illustrative overview how molecular properties, such as energies and forces, can be learned as a function of three-dimensional molecular structures. Therefore, a molecular database is required comprising atomic structures and quantum chemical reference calculations (1). A mathematical representation encodes the atomic configurations as input vectors (2). These representations can either represent the entire molecule (global) or encode local atomic environments within a cutoff value R_{cut}. The represented molecular structures serve as input values for the ML model. The model "learns" the structure-property relationship via regression (3). To this end, kernel-based methods or ANNs are commonly employed. This sketch lists in (3) the KRR method as this is used during the work of this thesis. The explained route (a) corresponds to the more "traditional" way of training a MLIP. With the advent of GNNs, atomic representation learning and the regression part is simultaneously done in one "learning" step (b). Furthermore, model parameters have to be adjusted and the model has to be validated (4). Finally, it can be applied in the respective task (5).*

3.1 The Database

A reference database $\mathcal{D}$ builds the basis for any ML applications in theoretical chemistry, as the ML algorithm learns from that database. The data set should be constructed to be a representative set of atomic configurations and respective high quality calculated reference properties, such as energies and forces. In many cases, these reference calculations are computed via DFT. However, depending on the specific task, for which an ML model is trained, the choice of the reference method has to be well considered, since it determines the accuracy of the model. [26, 31]

Over the last decade, researchers have established large QM-based molecular databases such as QM9 [55, 56], QM7-x [57], MD17 [75], OE62 [76], OC20 [11] or Rad-6 [1] for the development of ML methods and corresponding benchmark purposes. These databases contain either only equilibrium configurations or both equilibrium and non-equilibrium structures. QM9 or Rad-6 for instance, are databases consisting of equilibrium configurations. As a consequence, they are mainly usable for fitting relationships between equilibrium configurations and respective energies [1, 2]. Moreover, they are rather unpractical in the use of generating MLIPs applied in dynamical approaches. These databases do not completely cover the molecular phase space, since configurations between equilibrium structures are not included. Thus, the model has to "guess" function values in those regions. This may lead to instabilities and unpredictable situations in dynamical evaluations. As a consequence, further databases (e.g. MD17 or QM7-x) were developed that explicitly add samples from MD simulations or displaced configurations (along normal modes)

to account for that problem. Consequently, ML models can be trained and used for dynamical approaches [3].

For many applications, e.g. in the field of materials science, large QM-based databases are usually not available. Therefore, the researcher has to generate the reference database from scratch. These database generations are commonly done within an iterative procedure and the help of active learning. Here, the database grows in each iteration by sampling new configurations from atomistic simulations. These simulations are performed by the potential of the current iteration. New structures are selected by an active learning criterion (uncertainty estimate or similarity measure). Reference calculations are performed for these structures and they are added to the training set of the next iteration. As a result, the potential learns from itself about sparsely sampled or unstable phase space regions and appropriate databases are generated, thereby.

To decrease the amount of reference calculations, physical knowledge is directly incorporated into the ML model (see section 3.2 for details). Furthermore, lower level baseline methods can be used (Δ-machine learning, Δ-ML). This has proven to be particularly useful in applications, where highly accurate ML models are required and thus, very expensive reference methods are used [77]. In Δ-ML, the property of interest is evaluated with both levels of theory methods. The ML model will then learn the differences between these methods during the training. This results in a simplified and smoother learning problem. Including a baseline model consequently helps to be data-efficient and it additionally increases the robustness of the model. [26]

Finally, it remains to be clarified how a representative set of training configurations is drawn for a specific task, if a fixed database is available. Therefore, some kind of similarity measure is required that maximizes the variety of structures in the training set. Kernel functions incorporate such a metric, for which a more detailed explanation will be given in section 3.4. Let's assume for now that we have such a metric that measures the distances between data points in chemical space. Then, the **farthest-point sampling (FPS)** method can be employed to select a representative and diverse set of configurations from a database $\mathcal{D}$.

The FPS algorithm starts with a set of given structures $\mathcal{S}$. It is also possible that $\mathcal{S}$ contains only one structure at the beginning. In the next step, the algorithm selects a new structure A by maximizing the distances between A and all other structures in $\mathcal{S}$. Mathematically, this is:

$$A = \text{argmax}_{A \in \mathcal{D}} \left(\min_{A' \in \mathcal{S}} D(A, A') \right) \text{ with } D(A, A') = \sqrt{K(A, A) + K(A', A') - 2K(A, A')}, \quad (3.1)$$

with $D(A, A')$ being the distance between structure A and A' and $K(A, A')$ the respective kernel entry. The algorithm stops adding new structures to the training set when the chosen maximal number of structures are selected. [1, 35]. Note, that FPS can also be applied in iterative training workflows. [78]

3.2 Representations of Atomic Configurations

The previous section provides details about the database, from which the ML algorithm learns the atomic structure-property relationship. The consequent next step is therefore, an introduction of an atomic representation encoding the three-dimensional coordinates of the chemical system to make them accessible for the ML algorithm. Depending on the literature, molecular representations

are also denoted as descriptors or as molecular features. In general, such representations should fulfill a number of requirements to ensure a reliable and robust ML model. The following will list these requirements. [31]

Symmetry: A molecular representation should be invariant to translations, rotations and permutations of identical atoms, as the potential energy is also invariant to these symmetry operations. This consequently excludes the use of Cartesian coordinates as suitable representations for atomic configurations, since they are not relative values. Therefore, further developments of molecular representations have been made in recent years to fulfill the symmetry requirement. These additionally go beyond the simple use of internal coordinates, such as bonds, angles, or dihedrals. [74] Finally it should be noted here, that the ML model can in principle learn the invariances from the data. This, however, requires a massive amount of data and that is commonly not suitable in chemical applications. [79]

Uniqueness and generality: It is further intuitive that the same representation of a molecular structure should lead to the same energy. Moreover, the representation should be valid and applicable to all kinds of chemical systems. This includes both molecules and materials with periodic boundary conditions. [30]

Smoothness: In the generation of MLIPs, the potential energy is a continuous function in terms of atomic positions. Further, molecular forces can be derived as respective gradients of the potential energy with respect to the coordinates. This results in the requirement of smoothness for molecular representations. [74, 79]

Going beyond these physical requirements, Fig. 3.2 illustrates that representations can be constructed to either represent the entire molecular configuration within global representations, such as the Coulomb matrix [34], bag of bonds [80] and the many-body tensor representation (MBTR) [79] or in terms of local atomic environments defined within a certain cutoff value (local representations). Examples for the latter are the smooth overlap of atomic positions representation (SOAP) [81] or Behler-Parinello symmetry functions [36]. Global representations provide direct access to global properties such as the potential energy. However, their scaling to larger systems is rather poor and they are not inherently size-extensive. [31]

To overcome these limitations, Behler and Parrinello [36] introduced a representation, in which the molecule is not represented as a whole, but in terms of local atomic environments. Within this picture, the potential energy of a molecule A is then expressed as a sum of atomic energies ε_a

$$E(A) = \sum_{a}^{N} \varepsilon_a. \tag{3.2}$$

Respective representations encode the atomic environment by including information about the atoms neighborhood. Here, the neighborhood of an atom is usually defined within a certain cutoff value and comprises of different body order contributions. [31, 82]

It is worth to point out that the above mentioned assumption of locality is an approximation and is not verified by the SE. It has to be tested if this assumption is suitable for the specific task. [29] Nonetheless, when using local descriptions, long range contributions (e.g. electrostatics and dispersion) can be additionally incorporated into the ML model [38, 42, 83, 84].

Moreover, local descriptors naturally account for size-extensive predictions. Thus, they allow to reasonably predict larger systems compared to the ones in the training set. [74] A more detailed

explanation about size-extensive ML will be given in the respective sub-section of section 3.4.2.

Recently, GNNs [37–42, 85, 86] arrive as rather new kid to this family of machine learning potentials (see Fig. 3.2b). In contrast to the more traditional ML potentials, GNNs do not handle the representation and regression task separately, but combine them via message passing steps in one simultaneous learning task. Since this thesis contains an additional section about neural networks, further details about message passing networks will be given there. [82] For now, this section will give a closer look into two different representations as they are used during the work of this thesis—the local SOAP descriptor in [1] and the global MBTR in [2].

3.2.1 Smooth Overlap of Atomic Positions

The smooth overlap of atomic positions (SOAP) [81] representation is a local descriptor that encodes the atomic environment of an atom in any kind of chemical system. In the interests of simplification, we start with the construction of a SOAP representation for a chemical system that contains only one type of atomic species (e.g. for an atom in a C_{60} fullerene molecule). Afterwards, we will expand this derivation to multi-element systems in a second step.

The atomic environment $\mathcal{X}_a$ of the atom a is represented as the density in terms of atomic coordinates $\rho_a(\boldsymbol{R})$:

$$\rho_a(\boldsymbol{R}) = \sum_{b \in \mathcal{X}_a} \exp\left(-\frac{|\boldsymbol{R} - \boldsymbol{R}_{ab}|^2}{2\sigma_{\text{at}}^2}\right) g_{\text{cut}}(|\boldsymbol{R}|). \tag{3.3}$$

This density is defined as a sum of atom-centered Gaussians with variance σ_{at}^2. The sum ranges over all atoms b present in the atomic neighborhood of atom a within a cutoff radius. It should be explicitly noted that a as the central atom within the cutoff circle is also included in that sum. Furthermore, the function g_{cut} ensures a smooth decay of the density to zero outside the cutoff radius.

As discussed above, a representation should be invariant with respect to translations, rotations and permutations of identical species. Here, the constructed density is invariant to translations and permutations, however, not with respect to rotations. The latter can be additionally implemented by expanding the density in an orthogonal basis of radial basis functions $g_n(|\boldsymbol{R}|)$ and spherical harmonics $Y_{lm}(\hat{\boldsymbol{R}})$

$$\rho_a(\boldsymbol{R}) = \sum_{nlm} c_{nlm} g_n(|\boldsymbol{R}|) Y_{lm}(\hat{\boldsymbol{R}}), \tag{3.4}$$

and then using the expansion coefficients c_{nlm} for constructing a rotationally invariant representation $p_{n,n'l}(a)$ that is defined as the power spectrum of the density

$$p_{nn'l}(a) = \pi \sqrt{\frac{8}{2l+1}} \sum_m c_{nlm}^{\dagger} c_{n'lm}. \tag{3.5}$$

For practical reasons, the power spectrum is truncated after $n \leq n_{\text{max}}$ and $l \leq l_{\text{max}}$ contributions. Of course, the more contributions are included, i.e. larger n_{max} and l_{max}, the higher is the spatial resolution of the density and the more accurate is the resulting fit. However, as usual this comes with an increase in computational cost so that appropriate values have to be determined. [87, 88]

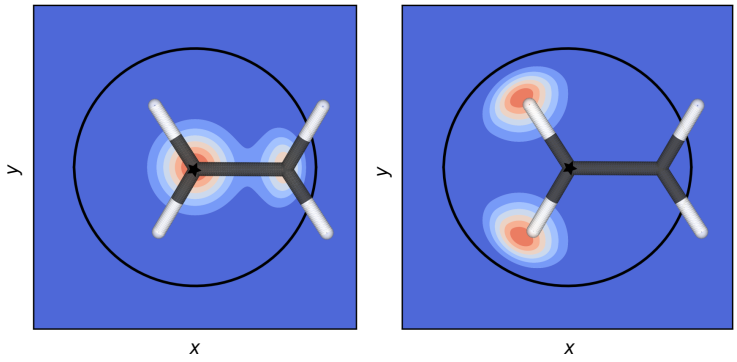


Fig. 3.3: Smooth Overlap of Atomic Positions. *Illustrative example of the neighborhood density function in an ethane molecule. Densities are constructed separately for each elemental species around a central carbon atom (indicated by an asterisk) within a cutoff radius (black circle). The left panel shows $\rho_C^C(\boldsymbol{R})$ and $\rho_C^H(\boldsymbol{R})$ is displayed on the right. This figure is a reprint from Ref. [1], published under the CC BY 4.0 license; http://creativecommons.org/licenses/by/4.0/. It is reproduced with permission from Springer Nature. Copyright ©2020, Sina Stocker, Gábor Csányi, Karsten Reuter and Johannes T. Margraf.*

To expand the SOAP representation for multi-element systems, separate densities are generated for each of the species Z individually

$$\rho_a^Z(\boldsymbol{R}) = \sum_{b \in \mathcal{X}_a^Z} \exp\left(-\frac{|\boldsymbol{R} - \boldsymbol{R}_{ab}|^2}{2\sigma_{\text{at}}^2}\right) g_{\text{cut}}(|\boldsymbol{R}|). \tag{3.6}$$

Figure. 3.3 illustrates such individually generated densities for an ethane molecule around a carbon atom (central atom) as an example. The individual densities are then again expanded in a basis of radial basis functions and spherical harmonics

$$\rho_a^Z(\boldsymbol{R}) = \sum_{nlm} c_{nlm}^Z g_n(|\boldsymbol{R}|) Y_{lm}(\hat{\boldsymbol{R}}). \tag{3.7}$$

The resulting partial power spectrum $p_{nn'l}^{Z_1Z_2}(a)$ contains now cross-species terms Z_1 and Z_2 and encodes the atomic environment of atom a

$$p_{nn'l}^{Z_1Z_2}(a) = \pi \sqrt{\frac{8}{2l+1}} \sum_m (c_{nlm}^{Z_1})^\dagger c_{n'lm}^{Z_2}. \tag{3.8}$$

In total, we obtain $\frac{N_Z(N_Z+1)}{2}$ vectors for a chemical system with N_Z atomic species, which are finally concatenated to one final SOAP representation vector $\boldsymbol{p}(a)$ for each atom. [88, 89]

3.2.2 Many-Body Tensor Representation

The many-body tensor representation (MBTR) [79] is an example for a global representation. In a more pictorial view, MBTR provides a fingerprint for a molecular geometry by returning the frequency of different many-body (k-body) contributions present in a chemical system. One-body terms are described as atom counts, two-body terms are represented as inverse interatomic

distances and angles are encoded as three-body contributions. Higher k-body contributions, such as dihedrals (four-body) may be included as requested. In many applications, however, MBTR is limited to the lowest two or three-body terms, while only one and two-body contributions have been used during the work of this thesis in [2].

Within MBTR, each element and k-body combination is smeared by a Gaussian and summed up to a final distribution function. In addition, a weighting function w_k is applied that smooths the k-body contributions to zero and ensures non-interacting particles at large distances. The individual k-body distribution functions are mathematically expressed as:

$$g_k(\boldsymbol{x}, z_1, ..., z_k) = \sum_{a_1,...,a_k}^{N} w_k(a_1, ..., a_k)\mathcal{N}(\boldsymbol{x}|G_k, \sigma_G^2) \prod_{b=1}^{k} \delta_{z_b, Z_{a_b}}, \tag{3.9}$$

with $z_1, ..., z_k$ being atomic numbers, G_k scalar k-body functions depending on the individual atoms $a_1, ..., a_k$, Z_a the proton number of an element and $\mathcal{N}(\boldsymbol{x}|\mu_G, \sigma_G^2)$ is a Gaussian distribution with mean μ_G and variance σ_G^2. The product of δ functions ensures the correct allocation of element combinations. As an illustrative example, Fig. 3.4 displays the resulting one and two-body distribution functions in ethane.

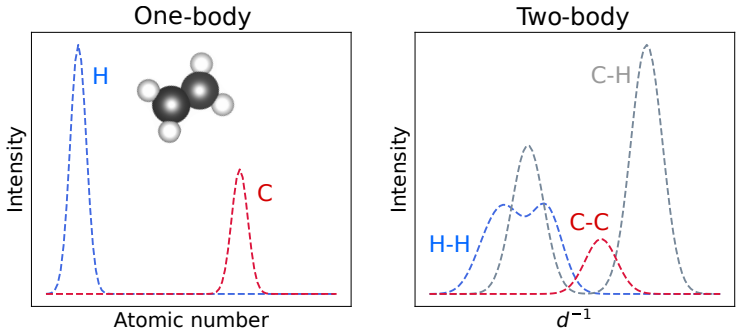


Fig. 3.4: Many-Body Tensor Representation. *The figure illustrates one-body and two-body distribution functions in the MBTR descriptor for ethane. Concatenating all k-body distribution functions results in the final MBTR feature vector.*

In the end, the global MBTR descriptor is a concatenated vector that consists of the discretized feature distribution functions for all elements and element combinations up to the chosen order, where symmetric contributions (i.e. H-C and C-H) are left out. In the example for the ethane molecule, where we have two one-body distribution functions and three two-body distributions the resulting vector is given as:

$$\boldsymbol{v}_{\text{MBTR}} = \boldsymbol{v}_{1,z_1} \oplus \boldsymbol{v}_{1,z_2} \oplus \boldsymbol{v}_{2,z_1z_1} \oplus \boldsymbol{v}_{2,z_1z_2} \oplus \boldsymbol{v}_{2,z_2z_2}, \tag{3.10}$$

where the first subscript denotes the k-body contribution and the second the elemental species or element contributions, respectively. $\oplus$ indicates concatenation.

During the work of this thesis in [2], a normalized version of MBTR was used to ensure size-extensive ML models and corresponding predictions. Therefore, each k-term distribution is normalized according to the corresponding k-body L^2-norm, to be not dominated by higher k-body contributions. The resulting representation for the ethane molecule has the following

form:

$$\boldsymbol{v}_{\text{iMBTR}} = \frac{\boldsymbol{v}_{1,z_1}}{|\boldsymbol{v}_1|} \oplus \frac{\boldsymbol{v}_{1,z_2}}{|\boldsymbol{v}_1|} \oplus \frac{\boldsymbol{v}_{2,z_1z_1}}{|\boldsymbol{v}_2|} \oplus \frac{\boldsymbol{v}_{2,z_2z_2}}{|\boldsymbol{v}_2|} \oplus \frac{\boldsymbol{v}_{2,z_1z_2}}{|\boldsymbol{v}_2|}, \quad \text{with } \boldsymbol{v}_1 = \boldsymbol{v}_{1,z_1} \oplus \boldsymbol{v}_{1,z_2} \text{ and } \boldsymbol{v}_2 = \boldsymbol{v}_{2,z_1z_1} \oplus \boldsymbol{v}_{2,z_1z_2} \oplus \boldsymbol{v}_{2,z_2z_2}. \tag{3.11}$$

This is particularly important for fitting intensive quantities. In [2], the intensive MBTR version was denoted as iMBTR. [2, 76, 79, 90, 91] For more details about size-extensive kernel-based ML see the respective size-extensivity subsection in section 3.4.2.

3.3 Universal Approximations and Loss Function

In computational chemistry, ML methods can be employed to learn a structure-energy relationship for chemical systems, i.e. having a surrogate model for approximately obtaining the potential energy of a given geometry. However, this structure-energy relationship can be arbitrarily complex and in many cases it is not solvable within a "normal" linear regression task. As a consequence, so-called universal approximators are required, which are able to imitate any continuous and smooth function with sufficient accuracy from a given training database. [26]

In many ML applications, this universal approximation of any relation between input and output data is realized by the concept of "linearization". This means, that the nonlinear problem in "real space" can be mapped to a so-called "feature space", where the problem can linearly be solved. In kernel-based ML methods, this is achieved by using the "kernel trick", which will be further discussed in section 3.4.1. Within neural networks (NNs), the nonlinear problem is divided into a linear and nonlinear part represented by connected layers of neurons. Here, the connections represent linear operations with adjustable parameters and the neurons are nonlinear activation functions. Specifically deep neural networks, where the network consists of many such layers, can learn any complex relationships between input and output data. [25] In this context, it is further interesting to mention that the equivalence of kernel-based regression and NNs can be shown. This holds for a NN with one hidden layer that is infinitely wide. [29]

Nonetheless, using universal approximators bears the risk of overfitting. In ML, overfitting denotes the capability of a function to perform very well on training data, while providing highly biased predictions on test data. Thus, it is not necessarily optimal to obtain an ML model that passes through each data point in the training set, but to have a model that **generalizes** to unseen data. Better generalization can be achieved by using simpler ML models. Consequently, regularization is often applied, which favors e.g. the selection of simpler models. [25, 26]

One common way to apply regularization is given by solving the following optimization problem

$$f = \arg\min_{f \in \mathcal{F}} \underbrace{\left[\sum_{i}^{M} \mathcal{L}\left(f(\boldsymbol{x}_i), y_i\right) + \lambda \mathcal{R}(\boldsymbol{\theta}) \right]}_{\mathcal{L}'(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y})}, \tag{3.12}$$

where the first part constitutes a loss function $\mathcal{L}$ that describes the differences between the model f and output values $\boldsymbol{y}$, usually as the squared error. The second part defines a regularization term $\mathcal{R}$, which is additionally added to the final regularized cost function $\mathcal{L}'$. Here, this $\mathcal{R}$ term depends on the model parameters $\boldsymbol{\theta}$ to shrink them to rather small values as this corresponds

to a simpler model. A common example is the Tikhonov regularization approach [92], which incorporates the L^2-norm of the model parameters to prevent the model from choosing values that are too large. In addition, the regularization parameter $\lambda > 0$ is a hyperparameter, that controls the complexity of the fitted model. It can be considered as a tuning parameter, with which the degree of generalization and accuracy can be defined. Besides the Tikhonov regularization also other regularization strategies, such as early stopping [93] or dropouts [94] can be employed, specifically for NNs. [26, 95, 96]

3.4 Kernel-Based Methods

3.4.1 The Kernel Trick

As indicated above, complex structure-energy relationships are usually nonlinear problems, which cannot be modeled with linear regression on the features of the input representation. To circumvent this limitation, nonlinear basis functions $\Phi(\boldsymbol{x})$ are introduced to map the inputs $\boldsymbol{x}$ into a high dimensional feature space as it is illustrated in Fig. 3.5. The crux, however, is to define such basis functions that are appropriate for the specific task. On the other hand, it can be computationally demanding to evaluate input values in the feature space, especially if it is high dimensional.

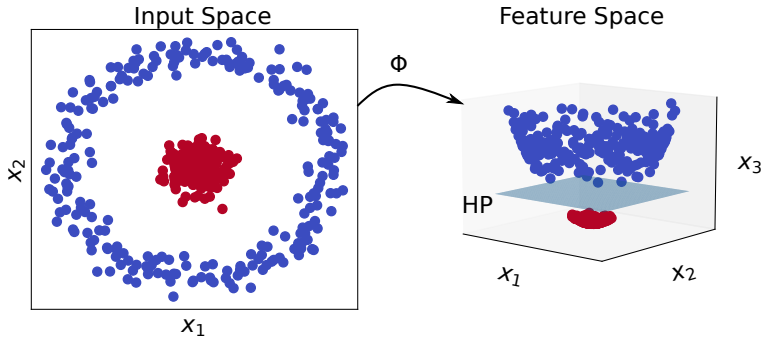


Fig. 3.5: The Kernel Trick. *The figure visualizes the basic idea of the kernel trick using a classification problem as an example. The left sub-panel shows a two-dimensional binary classification problem, which is not linearly separable (different colors define the two classes). By mapping the inputs into a higher dimensional feature space via the nonlinear basis function Φ the two classes can be separated linearly by a hyperplane (HP).*

On a more positive note, many ML approaches such as classification or regression problems can be reformulated in terms of scalar products between the input values. In case basis functions Φ are applied to the inputs, the reformulated problem does not depend on their specific definition anymore, but on the expression of their scalar product $\langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}') \rangle$. As a consequence, the explicit basis functions have in principle not to be known, however, the corresponding behaviour of the inner product for them. In this context, kernel functions may help, since they act on inputs in input space, however, perform as the scalar product in feature space. They are mathematically defined as:

$$k(\boldsymbol{x}, \boldsymbol{x}') := \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}') \rangle . \tag{3.13}$$

This can be understood as an introduction of a similarity measure, which defines how similar two inputs are in real space and consequently in feature space. [30] As a consequence, the explicit

basis functions have not to be designed, only a suitable kernel function has to be selected. An introduction to kernel functions that have been used during the work of this thesis in [1, 2] and resulting requirements for chemical applications are given in the following. For more details about kernel functions, the interest reader is referred to [97].

3.4.2 Kernel Functions

Prominent Kernel Functions

For two given inputs $\boldsymbol{x}$ and $\boldsymbol{x}'$ (e.g. two representation vectors from section 3.2) the following kernels are defined as: The **polynomial kernel**

$$k(\boldsymbol{x}, \boldsymbol{x}') = \left(\left\langle \boldsymbol{x}, \boldsymbol{x}' \right\rangle + \beta \right)^{\zeta}, \tag{3.14}$$

which can be transformed for $\beta = 0$ and $\zeta = 1$ into the **linear kernel**

$$k(\boldsymbol{x}, \boldsymbol{x}') = \left\langle \boldsymbol{x}, \boldsymbol{x}' \right\rangle. \tag{3.15}$$

The linear kernel performs as the original input features, without mapping them into a "higher" dimensional feature space. Besides these two, a further kernel function represents the **RBF** (radial basis function) or **Gaussian kernel**

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\gamma |\boldsymbol{x} - \boldsymbol{x}'|^2 \right), \tag{3.16}$$

where γ defines a length scale parameter. Using the RBF kernel corresponds to an infinitely large feature space. [30]

As it is shown in Eq. 3.1, kernels do not only provide a similarity measure between data points, but allow for explicitly calculating the distance between them. Furthermore, kernels contain additional parameters so-called hyperparameters, such as the length scale parameter γ in the RBF kernel. These parameters may have a drastic influence on the resulting fit if they are chosen incorrectly. The selection of appropriate hyperparameters will be further discussed in section 3.6.

Local and Global Kernels

Machine learning interatomic potentials typically employ the concept of locality, where the total energy is divided into a sum of atomic energy contributions (Eq 3.2) and respective atoms are encoded by local representations. However, DFT reference calculations for molecular systems only provide total energies and atomic energies are usually not available. As a consequence, local representations or corresponding local kernels have to be transformed into global kernel elements $K(A, A')$ measuring the similarity between two molecules A and A'. The global kernel matrix is then employed to fit on molecular energies.

Using local SOAP kernels [81]—SOAP representations in combination with polynomial kernels—as an example, two different global kernels can be constructed. One example is given by the **average kernel** [35, 89], which is mathematically expressed as

$$K(A, A') = \sum_{a \in A, a' \in A'} \frac{1}{N_A N_{A'}} k\left(\boldsymbol{p}(a), \boldsymbol{p}(a') \right). \tag{3.17}$$

Here, the sum runs over all atoms a and a' present in molecule A and A', respectively. N_A and $N_{A'}$ are the number of atoms in the molecules. Note that the average kernel (intensive kernel) is commonly normalized to ensure that the self-similarity is equal to one. A second example is given by the **sum kernel** [47, 98]

$$K(A, A') = \sum_{a \in A, a' \in A'} k\left(\boldsymbol{p}(a), \boldsymbol{p}(a')\right), \tag{3.18}$$

whereby this global kernel is simply expressed as the sum over all local kernel elements.

In summary, the main difference between these two kernels is that the average kernel is intensive, due to the averaging over all atoms. In contrast, the sum kernel is extensive and measures size differences between molecules. It has been demonstrated during the work of this thesis in [2], that ignoring size-extensivity in ML methods leads to poor transferability of ML models, when training on small molecules and predicting on larger configurations. Requirements for size-extensive kernels are given in the next sub-section. [1, 2]

Size-Extensivity

As kernel-based machine learning methods incorporate invariances with respect to symmetry operations, it would be consistent to also include size-extensivity in these methods. If the target property is extensive, like the atomization energy (AE) of a molecule, then the kernel element between molecule A and two non interacting molecules, should be twice as the self-similarity of A. This is mathematically expressed as:

$$K(A, 2A) = 2K(A, A). \tag{3.19}$$

On the other hand, when predicting intensive properties (e.g. AE per atom) the following condition has to be fulfilled:

$$K(A, 2A) = K(A, A) = 1. \tag{3.20}$$

As it was discussed during the work of this thesis in [2], both equations are not fulfilled, when using MBTR ($\boldsymbol{v}_{\text{MBTR}}$) in combination with the RBF (Gaussian) kernel. Since the Gaussian kernel includes the Euclidean distance in the exponent, the kernel element $K(A, 2A)$ will result in a value close to zero, when using the respective representation shown in Fig. 3.6a. The exact value for $K(A, 2A)$ will finally depend on the length scale parameter γ, however, it will not be equal to $2K(A, A)$. As a consequence, the MBTR/Gaussian kernel is not a size-extensive kernel. Equation 3.19 is fulfilled, when using MBTR in combination with the linear kernel. In contrast, the iMBTR in combination with the Gaussian kernel is an intensive kernel that fulfills Eq. 3.20. Using the respective representation visualized in Fig. 3.6b, the exponent in Eq. 3.16 will be equal to zero and the kernel element will consequently result in a value of one.

While the current section provides an overview about kernel functions and their requirements in chemical applications, there is still an open question of how we can use kernels to model a structure-energy relation for chemical systems. One possibility is to employ KRR, which is one type of regression, where model parameters can be obtained by solving a linear least squares problem. More details are provided in the next section. [2]

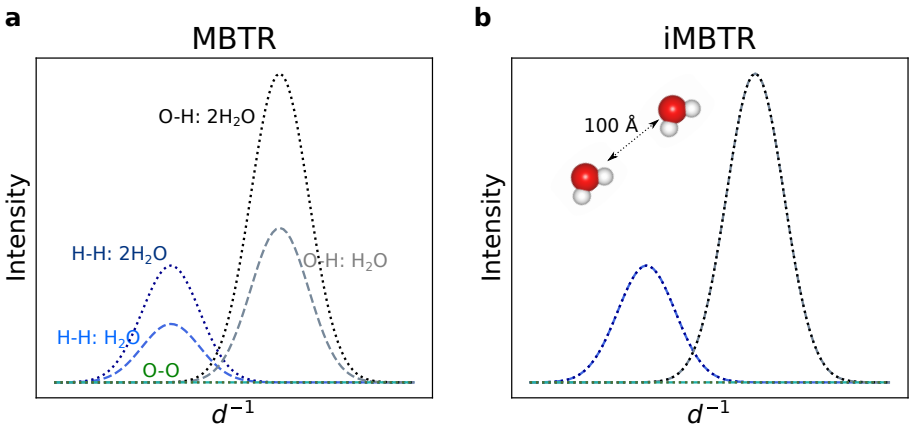


Fig. 3.6: Differences in MBTR and iMBTR for Size-Extensive ML. *The figure illustrates the differences in the two-body distribution functions for a single water molecule (dashed lines) and two non interacting water molecules (dotted lines) for MBTR (a) and iMBTR (b). Reproduced from Ref. [2].*

3.4.3 Kernel Ridge Regression

In kernel ridge regression (KRR), the target values $\boldsymbol{y}$ are approximated by a function $f(\boldsymbol{x})$, for which the following equation is fulfilled:

$$\boldsymbol{y} = f(\boldsymbol{x}) + \boldsymbol{\xi}. \tag{3.21}$$

Here, measurement noise in training data is allowed and incorporated via the $\boldsymbol{\xi}$ term. In case of fitting a MLIP, the target values are energies and corresponding input vectors are representations of the molecular geometries. According to the representer theorem [99], the learnable function $f(\boldsymbol{x})$ can be expressed in terms of weighted basis functions

$$f(\boldsymbol{x}) = \sum_{i}^{M} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i) \tag{3.22}$$

where the α_is are the expansion coefficients or model parameters ($\boldsymbol{\theta}$ in Eq. 3.12) of the ML method. The basis functions are kernels and their elements $K(\boldsymbol{x}, \boldsymbol{x}_i)$ measure the similarity between inputs $\boldsymbol{x}$ and $\boldsymbol{x_i}$ and act as their inner product in feature space. The sum in Eq. 3.22 runs over all M data points in the training set.

To obtain the function $f(\boldsymbol{x})$, the model parameters $\boldsymbol{\alpha}$ have to be determined by solving the optimization problem in Eq. 3.12 with the following regularized cost function

$$\mathcal{L}'(f(\boldsymbol{x}), \boldsymbol{y}) = \sum_{i}^{M} \left(f(\boldsymbol{x}_i) - y_i\right)^2 + \lambda \sum_{i,j}^{M} \alpha_i K(x_i, x_j) \alpha_j, \tag{3.23}$$

and rewritten in matrix vector notation

$$\mathcal{L}'(f(\boldsymbol{x}), \boldsymbol{y}) = (\mathbf{K}\boldsymbol{\alpha} - \boldsymbol{y})^T (\mathbf{K}\boldsymbol{\alpha} - \boldsymbol{y}) + \lambda \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha}. \tag{3.24}$$

Here, the Tikhonov regularization is applied where the regularization term is interwoven with the respective kernel elements to account for the correct norm in feature space. Solving the resulting

convex optimization problem provides a closed form solution for the coefficients $\boldsymbol{\alpha}$ and these can be obtained by setting the gradient to zero $\nabla_{\boldsymbol{\alpha}} \mathcal{L}(f(\boldsymbol{x}), \boldsymbol{y}) = 0$, which leads to the following solution:

$$\boldsymbol{\alpha} = (\mathbf{K} + \lambda \boldsymbol{I})^{-1} \boldsymbol{y}, \tag{3.25}$$

where $\boldsymbol{I}$ is the identity matrix. Equation 3.22 holds respective predictions for unseen data points $\tilde{\boldsymbol{x}}$. [1, 25, 27, 30]

It should be mentioned for completeness, that the here derived kernel-based regression method corresponds to the so-called weight-space view and is mostly denoted as KRR in literature. In contrast, the solution for the regression coefficients can be also derived from a probabilistic point of view. This is denoted as the so-called function-space view and is known as GPR. Gaussian process regression does not only provide predictions of unseen data points, but additionally allows for calculating corresponding uncertainties. [27]

Furthermore, KRR (or GPR) quickly becomes computationally intense especially for large training sets, since the inversion of the matrix in Eq. 3.25 scales as $\mathcal{O}(M^3)$. In this context, so-called sparse GPR is thus often applied. This method uses a (representative) subset of L configurations ($L << M$) to construct the model, while the loss function still uses information from the entire training set. This reduces the cost of sparse GPR to $\mathcal{O}(ML^2)$, which means that this method scales linearly with the number of training samples M. The prediction of a new data point scales in sparse GPR as $\mathcal{O}(L)$. [27, 29, 98] More details about sparsification are given in Ref. [100].

3.4.4 Kernel Principal Component Analysis

For many applications in computational chemistry, the underling data set is high dimensional. This could be problematic on the one hand, when the database is screened for certain patterns that are relevant for certain applications and no visualization is possible. On the other hand, the cost for computations might be artificially increased. Both drawbacks result from the fact that in many applications only a few number of features represent the process of interest, while remaining variables only include some noise. It might be thus very helpful to perform a dimensionality reduction of the data set (or so-called embedding), where the high dimensional data set is transformed into a lower dimensional space while important information is still retained. [26, 88]

This embedding can be achieved by the principal component analysis (PCA) tool or its corresponding non-linear counterpart kernel principal component analysis (kPCA). Since kPCA is used for visualization purposes during the work of this thesis in [1], here only the main concepts are introduced. This is done by giving an introduction to PCA. The conceptional idea of kPCA is the same and kPCA is derived by the kernelization of PCA. For mathematical derivations the reader is referred to [101, 102].

The main concept of PCA is to project the data set consisting of many variables (high dimensional) into a lower dimensional space. This new space is spanned by a set of orthogonal vectors: the principal components (PC). These PCs are linear combinations of the old variables and point into the directions of largest variance, since latent features are expected behind this variation. The PCs are obtained by solving the eigenvalue problems of the data set covariance matrix (for kPCA the kernel matrix). Importantly only the first few PCs that represent the highest proportion of the variance are chosen. Projecting then the data set onto these few PCs leads to the dimensionality reduction of the original data set. [101] Principal component analysis and kPCA are commonly

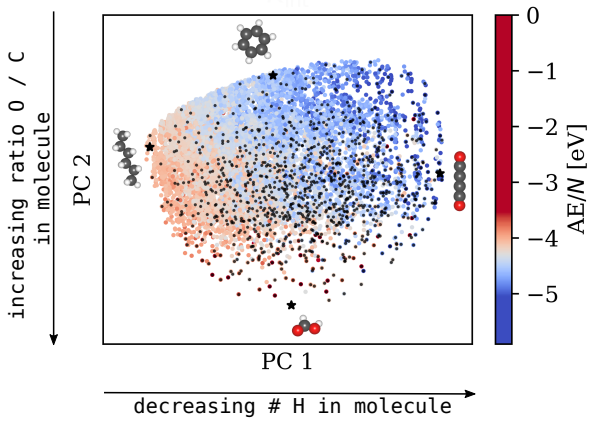


Fig. 3.7: Kernel Principal Component Analysis. *The graph exemplifies the visualization of molecular data sets via kPCA for the Rad-6 database. The arrows give an interpretation of the first two PCs that explain the underlying pattern. Black dots represent selected molecules that have been obtained from applying a FPS to the database. This figure is reprinted in parts from Ref. [1], published under the CC BY 4.0 license;* `http://creativecommons.org/licenses/by/4.0/`*. It is reproduced with permission from Springer Nature. Copyright ©2020, Sina Stocker, Gábor Csányi, Karsten Reuter and Johannes T. Margraf.*

used to visualize data sets. To this end, the different PCs are plotted against each other as it is shown in Fig. 3.7.

3.5 Neural Networks

As it was introduced in section 3.3, both kernel functions and neural networks (NNs) are universal approximators. Therefore, NNs can be used to learn complex structure-property relations to fit a MLIP for simulation applications. The most straightforward way to build a NN is to use an architecture, in which all input layer neurons $\boldsymbol{x}$ and a bias term $\boldsymbol{b}$ are fully connected to the output layer $\boldsymbol{y}$ via a weight matrix $\boldsymbol{W}$. This is mathematically expressed as:

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}, \tag{3.26}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are model parameters ($\boldsymbol{\theta}$ from Eq. 3.12) that are learned during the training. Nonetheless, Eq. 3.26 defines a function that linearly maps the inputs onto the output values and represents therefore not an universal approximator. To use NNs as universal approximators, at least one additional layer has to be inserted between the input and the output layer (see Fig. 3.8). This additional layer is denoted as hidden layer $\boldsymbol{h}$. Each neuron in the hidden layer performs a non-linear transformation on the received input via a non-linear activation function ϕ. The resulting output is expressed as

$$\boldsymbol{y} = \boldsymbol{W}'\boldsymbol{h} + \boldsymbol{b}', \text{with } \boldsymbol{h} = \phi(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}). \tag{3.27}$$

In many applications, however, deep neural networks are commonly applied, i.e. that not only

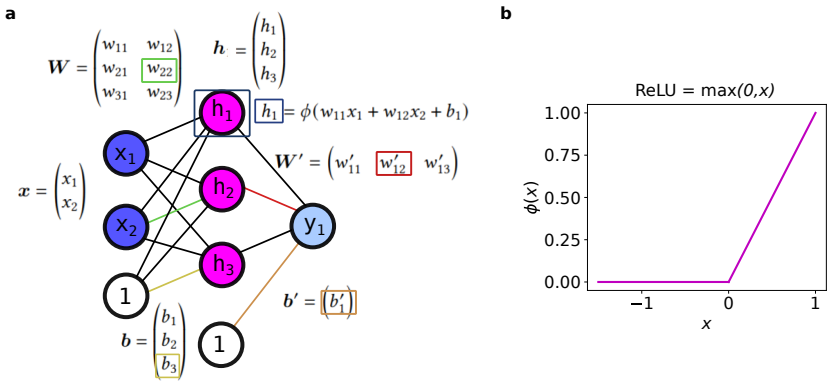


Fig. 3.8: Neural Networks. *The figure illustrates an example NN architecture with one hidden layer (a). In addition, an exemplary non-linear activation function $\phi(x)$ is shown in (b). Adapted from Ref. [25].*

one but many hidden layers are interconnected after each other. The output can therefore be rewritten as a nesting of all L layers before

$$\boldsymbol{y} = \boldsymbol{W}_{L+1}\phi(\boldsymbol{W}_L\phi(...\phi(\boldsymbol{W}_2\underbrace{\phi(\boldsymbol{W}_1\boldsymbol{x}+\boldsymbol{b}_1)}_{\boldsymbol{h}_1}+\boldsymbol{b}_2)...)+\boldsymbol{b}_L)+\boldsymbol{b}_{L+1}. \tag{3.28}$$

Note that the index labels have been changed here for simplicity. The number of used layers is a hyperparameter, which typically has to be defined by the user. In contrast, the weights and bias terms are ML model parameters and are determined during the training procedure within an optimization scheme. Similar to kernel methods, a (regularized) loss function is defined that quantifies the difference between predicted and target values. The weights are obtained by minimizing the loss function with the help of the stochastic gradient descent method or other gradient decent methods, such as mini-batching in combination with the backpropagation algorithm. Thereby, the weights are adjusted over different iterations by updating their values based on the negative gradient of the loss function, scaled by the learning rate η_n. The latter defines a step size for the update and is also a hyperparameter. Applying these iterative minimization schemes for training NN models is crucial, since no closed form solution exists for obtaining the weights as it does in kernel-based methods. For NNs, cost scalings for new predictions of new data points are linearly with the number of model parameters and independent of the number of training configurations. [25, 26]

Graph Neural Networks and Message Passing

Besides descriptor-based NN potentials [36, 45], Graph neural networks [37–42] (GNNs) have gained great popularity in recent years. Graph neural networks are powerful tools that act on graph data. It is therefore an obvious step to apply them for molecules (or other chemical systems) as the molecule itself can be represented as a graph. Such a molecular graph consists of atoms represented as nodes, and bonds or atomic interactions denoted as edges. In many GNN frameworks, the edges are defined between all atom pairs within a cutoff radius. A special feature of GNNs are message passing (MP) iterations [59], through which atomic representations can be

learned from data. Consequently, no molecular representations with hand-crafted features have to be designed as it is the case in "conventional" NNs and kernel-based methods.

Within a message passing neural network (MPNN) framework, initial atom-wise vector representations $\boldsymbol{h}_a^{(0)}$ are assigned to the nodes (see Fig. 3.9a). These vectors capture the information of elemental species and are usually called "embeddings" as element-dependent knowledge is embedded in an high dimensional feature space. Furthermore, feature vectors are allocated with the edges $\boldsymbol{e}_{a,b}$, typically based on information about the interatomic distances. The atom embeddings are then refined though MP steps. Specifically, this means that in each MP step each node receives information from edges and nodes located in the neighborhood of the atom. To update the embedding of node a at iteration l the message $\boldsymbol{m}_a^{(l)}$ is calculated in terms of nodes and edges in the neighborhood ($\mathcal{N}$) and then combined with information from the old state $\boldsymbol{h}_a^{(l)}$ via the following equation

$$\boldsymbol{h}_a^{(l+1)} = u(\boldsymbol{h}_a^{(l)}, \boldsymbol{m}_a^{(l)}), \text{ with } \boldsymbol{m}_a^{(l)} = \sum_{b \in \mathcal{N}(a)} m(\boldsymbol{h}_a^{(l)}, \boldsymbol{e}_{a,b}, \boldsymbol{h}_b^{(l)}). \tag{3.29}$$

Here u is an update function and m models the interaction between the atoms. Both are learnable functions, usually represented as NNs. [38, 40, 103] Message passing neural network potentials typically vary in different implementations of the update and interaction function. [25] After T MP steps, final atom embeddings $\boldsymbol{h}_a^{(T)}$ are obtained and fed into an additional network that is trained to perform the regression task and that predicts atomic energies. These are pooled together to the final total energy of a molecular configuration. [25, 26, 38, 50, 103, 104]

Although GNNs use cutoff functions similar to the SOAP representation, the resulting embeddings can encode information from beyond as it is visualized in Fig. 3.9b. In the first MP step, each node receives information from its neighbors within the cutoff sphere. Thus, each updated node contains both its own information and information from its surroundings. In the next MP step, nodes indirectly receive some information beyond their cutoff, since the neighboring atoms already received information from their neighbors within their cutoffs. This procedure commonly ends after 3-6 MP steps. [25, 103]

Similar to representations in kernel-based methods and descriptor-based NNs, the here learned representations should be invariant (or equivariant) to symmetry operations. The invariance with respect to permutations is fulfilled due to the sum operation in Eq. 3.29. Rotational invariance is obtained in most GNNs such as SchNet [37] by using pairwise distances. Furthermore, directional message passing networks allow leveraging higher-order contributions like angular information. [25, 82] A state-of-the-art representative of the latter is the geometric message passing neural network (GemNet) [41], which has also been used during the work of this thesis in [3].

As in traditional MPNNs, GemNet assigns initial embeddings to all atoms ($\boldsymbol{h}_a^{(0)}$). Additionally, GemNet embeds the interactions between atoms based on atom pairs within a cutoff radius. These pairs are represented via **directed** edge embeddings $\boldsymbol{e}_{a,b}^{(0)}$. These edges thereby define a direction in 3D, where it points from atom a to atom b. This direction allows to define angles between triplets of atoms φ_{abc} and dihedrals θ_{abcd} between quadruplets of atoms. The atom and edge embeddings are then updated via several MP steps based on other atom and edge embeddings within the cutoff sphere. These MP steps additionally incorporate information about the distances, angles and dihedrals of the respective atoms. After each MP step, the model outputs a separate set of atom and edge embeddings, which are transformed into local energy contributions. These contributions are then summed up to obtain the molecular energy. In addition, GemNet can predict forces by

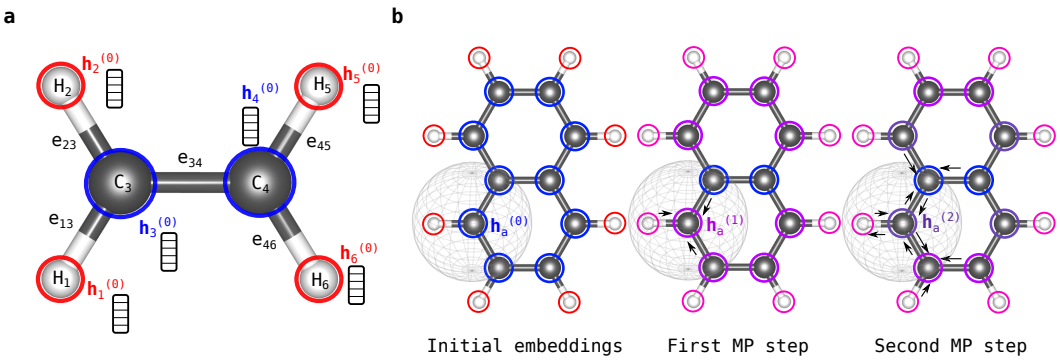


Fig. 3.9: Message Passing in Graph Neural Networks. *(a) Illustration of the initial node embedding in an ethane molecule. For simplicity, the cutoff sphere has been omitted and edge features $\boldsymbol{e}_{a,b}$ are allocated between direct neighbors. (b) Visualization of MP steps for a more complex naphthalene molecule. In the first MP step, node a receives information (black arrows) from neighboring atoms within the cutoff sphere (gray circle), i.e. that $\boldsymbol{h}_a^{(1)}$ gets updated based on its own and neighbors information. Here, the colors represent a measure to visualize the information exchange of updated embeddings. In the second MP step, node a indirectly receives information beyond the cutoff. Adapted from Ref. [103].*

calculating the negative energy gradients with respect to the atomic positions. For more details about GemNet and directional message passing the interested reader is referred to [40, 41, 85].

3.6 Hyperparameter Selection

As already seen in previous sections, ML models contain additional parameters, so-called hyperparameters, that have to be defined before the final ML model is trained. These parameters may have a drastic impact on model generalization, if they are wrongly selected. In general, the model itself can be considered as a hyperparameter, i.e. the choice of which kernel function is employed or which NN architecture is used. Apart from the model itself another type of hyperparameter exists, given by additional model parameters that are not determined within the training procedure itself. To this type belong for example the regularization parameter λ, the length scale parameter γ of the RBF kernel or the learning rate η_n in a NN. Appropriate values for these parameters have to be determined to ensure a decent model generalization error. [25, 26]

In many cases, hyperparameter selection is done within a grid search, an optimization procedure or both in combination with a validation set. Furthermore, Bayesian approaches [105] can also be applied. When using a grid search for the hyperparameter selection, the training set is divided into two subsets, where the larger one defines the training set and the smaller one a validation set. The hyperparameters are then systematically varied and the respective error (in many cases the root mean square error) is evaluated. Finally, those hyperparameters are typically chosen that perform best on the validation set.

Since the choice of the validation set may influence the hyperparameter selection, similar approaches exists with focus on improving the statistics. A prominent example is the k-fold cross-validation, where the training set is divided into k subsets and one of these subsets represents the validation set. The remaining $k-1$ subsets represent the training set. The training is then repeated k times, while using always a different subset of the k folds as validation set and the rest for training. The final hyperparameters are chosen from the ensemble of k models. Note, that each training is

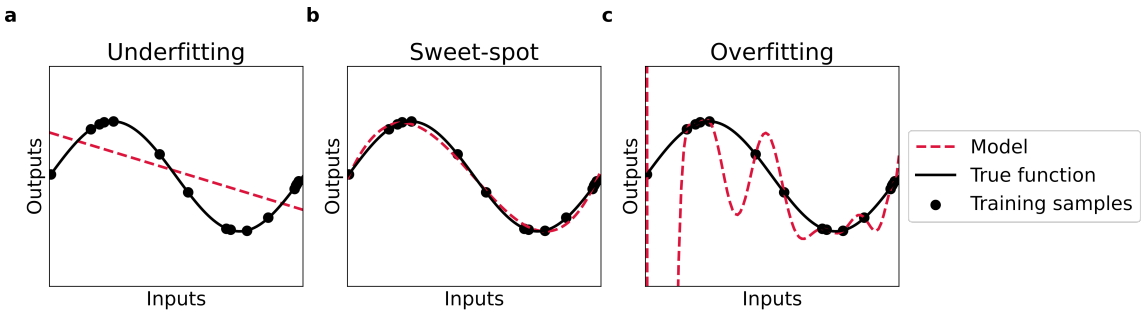


Fig. 3.10: Model Complexity. *The figure illustrates different situations for bias and variance trade-offs. Therefore, ML models of different complexity are displayed as red dashed lines in each sub-panel.(a) The ML model is too simple to describe the true function (underfitting). (b) The ML model is of reasonable complexity to provide an appropriate description of the true function. (c) The ML model is too complex to describe the true function (overfitting).*

combined with a hyperparameter selection part similar as described for a single validation set before. It should also be mentioned here that for many applications the hyperparameters are quite robust and usable for different databases. Moreover, some hyperparameters have physical meanings or can be derived as educated guesses from the data.

After successful hyperparameter selection, the ML model is then fitted on the training set and predictions are evaluated on a test set. The latter was already separated from the training set before hyperparameter selection. The resulting test set error is used to approximate the generalization error. [25, 26]

In this context, the terms bias and variance are often used. The bias describes the error between the predictions and the true values. A high bias typically means that the model is highly regularized and the prediction error for both the training and the test set are large. This is also known as underfitting, which means that the resulting function is too simple (see Fig. 3.10a). On the other hand the variance describes the ability of the model to generalize to unknown data. High variance means that the function is usually too complex. A model that has high variance and low bias is mostly overfitted, i.e. that the training data produce a tiny error, while the test set error is huge (see Fig. 3.10c). For a good ML model a sweet-spot between the bias and variance has to be determined, which is illustrated in Fig. 3.10b. Consequently, this leads to low training set errors and good generalization to unseen samples. [26, 106]

4 Publications

This chapter gives an overview about the published articles that build the basis of this cumulative thesis. Summaries of each article as well as individual author contributions are provided. The articles themselves and supplementary information are attached as appendix.

4.1 Machine Learning in Chemical Reaction Space

Sina Stocker, Gábor Csányi, Karsten Reuter and Johannes T. Margraf
Nat. Commun. 11, 5505 (2020).
DOI: https://doi.org/10.1038/s41467-020-19267-x

Summary

In this paper, we used established chemical compound space ML methods and adjust them for use in reaction energy predictions in reaction space. Existing databases utilized for the development of ML models in chemical space cannot be used for this purpose, as they only consist of closed-shell molecules and thus lack important intermediates and fragments. Therefore, as an indispensable first step towards the exploration of reaction space, an exhaustive DFT-based database (Rad-6) and corresponding reaction network (Rad-6-RE) of open and closed-shell systems have to be generated to describing bond dissociation and formation events in chemical reactions.

For the ML models, we employed the SOAP representation and constructed the sum and average kernels as introduced in section 3.2 and 3.4. Kernel ridge regression was used to predict atomization energies (AE) of molecules from the Rad-6 database and calculate reaction energies (RE) in Rad-6-RE. Furthermore, we applied the FPS method (section 3.1) to select a representative training set of molecular configurations based on both, the sum and the average kernel. Our results exhibited good performances and similar AE prediction accuracies for both kernels and training set selections. However, we noticed significant differences in the calculations of corresponding RE in Rad-6-RE. Specifically, for the models based on the extensive FPS training set selection we obtained huge RE errors for small training set sizes. These discrepancies could be attributed to essential hub molecules that are involved in many reactions in the network. These molecules should be included in the training as an incorrect prediction of their AE increases drastically the RE error. Appropriate weightings in the loss function and training set selections according to the network connectivity can solve this issue.

All described models used the optimized DFT geometry as inputs to predict the respective DFT energies. This constellation of input-output values is rather impractical for generating ML models in practice. If the relaxed DFT geometry were known, the corresponding energy would also be available and the ML model redundant. Therefore, we further generated ML models, which use force field (UFF) [19] based geometries as inputs to predict the DFT energy (of the DFT minimum geometry) as target value. These ML models show the same trends as the pure DFT models, however, with higher mean absolute errors.

Finally, we employed the predicted reaction energies to reduce a complete reaction network with more than 21,000 reactions to the essential steps with methane combustion as an example. Therefore, a micro-kinetic simulation was used and the reaction energies were predicted via the largest UFF-based ML model with the average kernel and intensive FPS. With this simulation, we were able to reduce the network to a total of 887 reactions containing all reaction intermediates proposed in literature. Moreover, we found unexpected intermediates that have not been considered before. These findings illustrate the importance of ML methods to rationally and non-empirically reduce reaction networks, as essential steps could be otherwise overlooked when constructing these networks (simply) by chemical intuition.

Individual contributions

The idea of this paper is the consequent advancement of the systematic enumeration of molecules and resulting reaction networks from Johannes T. Margraf and Karsten Reuter [13] to offer a route for the reduction of such exhaustive networks to the most relevant sub-parts. Therefore, Johannes T. Margraf provided the UFF-based geometries of the Rad-6 database and the Rad-6-RE network based on the algorithm developed in [13]. He further performed the micro-kinetic simulation and corresponding analysis as well as the BS-DFT single point calculations. All authors were involved in devising the project. Gábor Csányi especially contributed with his knowledge about the SOAP representation and corresponding kernels.

I performed all DFT geometry optimizations and DFT single point calculations on UFF geometries for the Rad-6 database in a high throughput fashion. Moreover, I analyzed the respective DFT optimization outcomes regarding the consistency with initial UFF geometries to ensure that the same molecular topology is used in the force field based ML models and consequently build up the final Rad-6 database. I further fitted all ML models including both DFT-based and UFF-based models, implemented the sum kernel in the `mltools` [107] package, carried out the FPS for training set selections and analyzed the results on the basis of self written python codes and scripts. Moreover, I did the kPCA for the visualization of the Rad-6 database, performed the RE analysis on Rad-6-RE and created respective figures for the paper. I drafted the initial version of the manuscript, which was revised and edited by Johannes T. Margraf. The final draft was proofread and refined by Gábor Csányi and Karsten Reuter.

4.2 Size-Extensive Molecular Machine Learning with Global Representations

Hyunwook Jung*, Sina Stocker*, Christian Kunkel, Harald Oberhofer, Byungchan Han, Karsten Reuter and Johannes T. Margraf
ChemSystemsChem 2, e1900052 (2020).
DOI: https://doi.org/10.1002/syst.201900052

Summary

Machine learning models accelerate the exploration of chemical and compound space or can be used to study for example large reaction networks. To train such models, suitable representations have to be developed that encode the molecular geometry. Such representations are typically categorized into global and local representations, for which the differences are shown in section 3.2. While many ML models are based on local representations, for which size-extensivity is by construction fulfilled, the correct treatment of differently sized molecular geometries with global representations is not always ensured.

In this paper, we showed that this limitation can be overcome in kernel-based ML, using the many-body tensor representation (MBTR) as an example. We stated that the original version of MBTR together with a Gaussian kernel (MBTR/Gaussian) does not fulfill the size-extensivity requirements as shown in section 3.4.2. However, MBTR is extensive when used with the linear kernel (MBTR/linear). Furthermore, we introduced a normalized version of the MBTR (iMBTR) (see section 3.2 for details), which ensures size-extensivity in combination with the Gaussian kernel (iMBTR/Gaussian).

The importance of size-extensivity was demonstrated by training the ML models on small molecules and evaluating predictions on significant larger systems. This test study showed that size-extensive models such as iMBTR/Gaussian and MBTR/linear generated energy predictions that show good correlations with DFT reference calculations. On the other hand, we found that MBTR/Gaussian performs adequately when training and test molecules are of similar sizes but produced disastrous results when extrapolating to larger systems. Our findings substantiate the significance of size-extensivity for building transferable and data-efficient ML models.

Individual contributions

The idea behind this paper was developed in parallel to my work on "Machine Learning in Chemical Reaction Space" [1]. Since we found size-extensivity to be essential for a proper description of reaction energies, the latter project focused on the local SOAP representation, which (unlike global representations) naturally ensures this property. This led Johannes T. Margraf and me to explore ideas for size-extensive ML models with global representations. In this context, I developed a range of ML models for molecular systems based on MBTR in combination with different kernel functions. This allowed me to connect the lacking size-extensivity in the MBTR/Gaussian kernel with the normalization conditions of the representation, kernel and fitting target. As a consequence, Johannes T. Margraf and I designed an improved version of MBTR that yields the correct normalization and is size-extensive.

*These authors contributed equally

To demonstrate the benefits of this new methodology, we decided to apply it in a setting that underscores the importance of size-extensivity, namely training on small molecules and predicting the properties of large ones. This application was realized by Hyunwook Jung under my supervision, during his research stay at TUM. There, he trained the corresponding ML models, created the figures for the paper and analyzed the results together with me and Johannes T. Margraf. Christian Kunkel recalculated the energies for the QM9 database with the same DFT settings used for the OE62 data set to ensure consistency of the reference calculations.

I was involved in drafting the manuscript and writing the theory section, where we introduced the requirements for kernels to be size-extensive. Furthermore, I created the table of content figure. All authors contributed to writing the manuscript.

4.3 How Robust are Modern Graph Neural Network Potentials in Long and Hot Molecular Dynamics Simulations?

Sina Stocker*, Johannes Gasteiger*, Florian Becker, Stephan Günnemann and Johannes T. Margraf
Published on ChemRxiv. Cambridge: Cambridge Open Engage, (2022).
DOI: https://doi.org/10.26434/chemrxiv-2022-mc4gb‡
Submitted for publication to Machine Learning: Science and Technology.§

Summary

Graph neural networks (GNNs) are powerful tools to accurately predict molecular properties on a QM accuracy level. Their excellent performance has been widely demonstrated on established benchmark databases, however, their robustness in real chemical simulations has been less explored. In this paper, we wanted to make up this omission and test the applicability of GemNet [41]—a highly accurate GNN—in long and hot MD simulations as an example. To this end, we trained several GemNet models with different training set sizes drawn from the QM7-x database [57] and ran a total of 245 ns of MD trajectories. Our results demonstrated that models with low test set errors do not automatically produce stable MD trajectories. In contrast, highly stable dynamics could be generated with exhaustive GemNet models, i.e. with sufficiently large training sets of 300,000 configurations and more. This is notable, since stable potentials are commonly generated with iterative learning procedures, e.g. in the field of kernel-based ML. Such training procedures are more difficult to enforce with complex GNN architectures as they typically require more training data and times.

Furthermore, we used the most exhaustively trained potential with a training set size of 3.2 Mio molecules to extrapolate predictions for significantly larger molecules and explore the robustness in high-temperature MD simulations. The resulting stable trajectories show systematic errors for energy predictions when validated with DFT, however, high correlations and impressive accurate force predictions. We additionally made the 3.2 Mio GemNet potential and our corresponding python wrapper for performing MD simulations with the `atomic simulation environment` (ase) [108] package publicly available.

Individual contributions

The paper came about as part of the interdisciplinary IGSSE project between the chair of theoretical chemistry and the informatics department "Data Mining and Analytics" at TUM with Johannes T. Margraf as project team leader (PLT). Johannes Gasteiger is a PhD student of the latter department and is working on GNNs as cost-effective alternative of QM methods. He developed the theory and components of the GemNet architecture [41] as well as for its predecessor DimNet++ [85]. The idea of the current contribution was developed by Johannes T. Margraf to combine the knowledge of both research fields, namely using GemNet and explore its robustness in MD simulations as these dynamical tests are missing in literature. To this end, Florian Becker fitted

*These authors contributed equally
‡This content is a preprint.
§An updated version of the peer-reviewed and accepted paper is now publicly available on DOI: https://doi.org/10.1088/2632-2153/ac9955.

the GNN potentials in the learning curves of the paper and provided the ase wrapper during his Master thesis, while he was under the direct supervision of Johannes Gasteiger.
I was responsible for the chemistry part of the paper and performed the MD simulations as well as respective DFT calculations for validating the dynamics. Furthermore, I carried out the respective analysis of the results regarding the MD simulations and prepared all figures in the paper. I drafted the manuscript and Johannes Gasteiger wrote the respective parts about the GemNet architecture. The draft was revised by Johannes T. Margraf and proofread by Stephan Günnemann.

5 Conclusions and Outlook

Over the last decades, ML and AI have emerged as powerful tools that have become an integral part of our daily life, which ranges from smart items and voice assistants up to self-driving transport vehicles. Furthermore, these statistical methods are increasingly used in the field of computational chemistry to provide new insights into chemical processes by accelerating physical property calculations at a fraction of the cost in contrast to the underling QM-based computations. By this, ML models allow for the simulation of chemical systems at desired length and time scales, thus bridging the gap between accuracy and efficiency that has existed since then between QM-based methods and empirical force fields. As an essential first step towards the successful generation of ML models, a data set is needed, from which the ML model can learn the structure-property relationship of interest. Generating molecular databases that cover large parts of of the chemical space is computationally intense as the QM reference calculations are expensive. Therefore, ML concepts that have been developed on established databases in compound space should be transferred to further sub-parts of the chemical compound or reaction space in order to solve various chemical questions in data-efficient approaches.

This cumulative thesis has aimed to make a contribution to this field by exploring the transferability of chemical ML. This was done by starting to explore the applicability of established compound space ML methods in chemical reaction space. Therefore, an appropriate database (Rad-6) had to be generated that contains molecular and radical species occurring in large and complex reaction networks. We used the FPS algorithm to select representative training sets and employed KRR with the sum and average kernel to predict AE and additionally calculate corresponding RE in the Rad-6-RE network. We found that some molecules are key for the assessment of RE predictions as they are involved in many reactions and therefore may drastically influence the error. Consequently adjustments had to be made. The predicted RE could be used for the reduction of large and complex reaction networks.

Besides the exploration of chemical reaction space, we further investigated the transferability of kernel-based ML models to predict molecules with large size differences. This is particularly important for ML with global representations as these are per se not size-extensive. We showed how size-extensivity can be included into the MBTR by using appropriate normalization conditions and kernel functions. In addition, we demonstrated the relevance of size-extensivity by designing a difficult test case and training ML models on small molecules while predicting much larger molecules. Our results showed that size-extensive ML models are necessary to accomplish this highly challenging task. However, for an adequate and data-efficient description of large molecules, some of them should be included in the training set. Otherwise phenomena like long range interactions cannot be described properly.

Finally, we explored the robustness of modern GNN MLIPs (here for GemNet) in molecular dynamics simulations, when trained on QM7-x. Their complex architecture and the resulting long training times make it difficult to apply iterative training, which has proven to be particularly effective for generating stable MLIPs. Therefore, large databases with many molecular configurations such as QM7-x are very important for generating stable potentials. Nevertheless,

GemNet can be considered as a cost-effective option to QM-based methods, providing impressively accurate results when trained on sufficiently large training sets.

The combination of machine learning and computational chemistry is an emerging field, in which great progress has been achieved in recent years. Nevertheless, further developments and improvements are still necessary. Specifically, in the field of catalysis I expect great progress for the design of new catalysts. The continuous further development of new machine learning concepts is particularly key to this field, since the tremendously large search space for new catalysts cannot be explored otherwise. Machine learning models are required that allow for the simulation of large length and time scales and thus enable the prediction of thermodynamic and kinetic data, the exploration of reaction networks and an acceleration of the determination of transition states (TS). In this context, researchers may develop MLIPs for surface science that can predict reaction barriers in large reaction networks. This includes not only TS searches within low-coverage regimes, but also dynamical approaches at high coverages. Specifically, the latter is important to model scenarios, which are much closer to experiments and therefore allow for more realistic descriptions.

Furthermore, including long range interactions into ML models is also a major topic at the moment. Machine learning interatomic potentials are mostly built with local representations, which give good results for many applications. However, long range effects, such as dispersion or electrostatics, may play a significant role in the transferability and scalability of ML models. Therefore, they should be included if requested.

Last but not least, the sufficient production of appropriate data is always important for fitting ML models. As mentioned above, the generation of accurate reference data in computational chemistry is very expensive and therefore solutions are required. One way to build data-efficient ML models is to use physical baseline methods. These methods already describe large parts of the physics and consequently only the differences between the QM and baseline method has to be learned. This typically produces accurate and robust potentials. On the other hand, active learning in combination with uncertainty estimates additionally supports data-efficient approaches. Thereby, the machine learning model learns by itself, which data should be included in the training process. Even though these two approaches help to build data-efficient ML models and reduce the number of reference calculations, it is still inevitable to create and publish sophisticated and large databases in the field of theoretical chemistry. In addition, much more experimental data should be used for training ML models. The collaboration of theoretical chemists with experimentalists and data scientists will be essential in the future for the successful generation of even better ML models. This underlines the idea of chemistry as a central science where many different disciplines are interconnected.

Acknowledgments / Danksagung

First of all, I would like to thank my supervisors Prof. Dr. Karsten Reuter and Dr. Johannes Margraf for giving me the opportunity to do this work. Specifically, I would like to thank Dr. Johannes Margraf for the scientific discussions, the support—whenever needed—and also for your trust. You really became a mentor for me and I have learned a lot, even beyond scientific aspects. Many thanks also to Prof. Dr. Karsten Reuter for the strategic support, the scientific advice and the opportunity to finish my PhD at the FHI in Berlin. I would also like to thank Prof. Dr. Gábor Csányi for the support and ongoing discussions about machine learning. Thank you very much for hosting me during my research stay in Cambridge.

Furthermore, a big thank you to the whole group in Berlin and also Munich for the many ongoing discussions about science, but also for the many group events and outings we had together (Oktoberfest, sailing trips to the Ammersee and international food evenings). Many thanks to Martin, Simon, Thorben, Hanna, Zausi, Carsten, Frederic and many more for the beer evenings we had not only in person, but also virtually during the pandemic and for the great time, we spent together doing sports. Thanks also to the group workshop organizers and to the whole IT-team. You all do a great job. Furthermore, a big thank you goes to the secretaries Julia Pach and Ruth Mösch for the help and advises in bureaucratic matters. I would also like to thank our Margraf subgroup for the nice working atmosphere and our subgroup outings like the Chinese cocking events. Moreover, it was a great pleasure to participate in the Berlin Marathon with our running group. Simon, Thorben, we can be really proud of ourselves. Many thanks also to Arobendo for the evenings we spent together playing games and the introduction to the Berlin life. I would like to thank my co-authors, especially Hyunwook, Johannes, Florian and Christian for the hard work we put into the papers and for scientific discussions. Furthermore, I gratefully acknowledge financial support from IGSSE and the FHI. In addition, I thank my former lecturer Prof. Dr. Dirk Flottmann for the advises regarding career planning and I am very happy that we still keep in touch.

Großer Dank gilt natürlich auch meiner Familie, speziell meinen Eltern und meiner Schwester. Danke für eure permanete Unterstützung, eure Geduld und dass ihr immer an mich glaubt. Vielen Dank auch an meine zukünftigen Schwiegerelten für eure Unterstützung und eure Besuche in München und Berlin. Mein größter Dank gehört jedoch dir, Simon, für die nun fast 9 Jahre, in denen wir unseren Weg gemeinsam beschreiten. Ich danke dir für deine Liebe, dein Vertrauen und deine Geduld. Es ist sehr schön, dass du immer zu mir stehst, egal ob in guten oder in stressigen Zeiten. Ich bin sehr dankbar darüber dich immer an meiner Seite zu wissen.

Bibliography

[1] S. Stocker, G. Csányi, K. Reuter, and J. T. Margraf, *Machine learning in chemical reaction space*, Nat. Commun. **11**, 5505 (2020).

[2] H. Jung, S. Stocker, C. Kunkel, H. Oberhofer, B. Han, K. Reuter, and J. T. Margraf, *Size-Extensive Molecular Machine Learning with Global Representations*, ChemSystemsChem **2**, e1900052 (2020).

[3] S. Stocker, J. Gasteiger, F. Becker, S. Günnemann, and J. T. Margraf, *How Robust are Modern Graph Neural Network Potentials in Long and Hot Molecular Dynamics Simulations?* ChemRxiv. Cambridge: Cambridge Open Engage, preprint, DOI: 10.26434/chemrxiv-2022-mc4gb, 2022.

[4] M. I. Zimmerman, J. R. Porter, M. D. Ward, S. Singh, N. Vithani, A. Meller, U. L. Mallimadugula, C. E. Kuhn, J. H. Borowsky, R. P. Wiewiora, M. F. D. Hurley, A. M. Harbison, C. A. Fogarty, J. E. Coffland, E. Fadda, V. A. Voelz, J. D. Chodera, and G. R. Bowman, *SARS-CoV-2 simulations go exascale to predict dramatic spike opening and cryptic pockets across the proteome*, Nat. Chem. **13**, 651 (2021).

[5] J. L. Medina-Franco, N. Sánchez-Cruz, E. López-López, and B. I. Díaz-Eufracio, *Progress on open chemoinformatic tools for expanding and exploring the chemical space*, J. Comput. Aided Mol. Des., 10.1007/s10822-021-00399-1 (2021).

[6] C. Kunkel, J. T. Margraf, K. Chen, H. Oberhofer, and K. Reuter, *Active discovery of organic semiconductors*, Nat. Commun. **12**, 2422 (2021).

[7] M. Gore and U. B. Jagtap, *Computational Drug Discovery and Design* (Springer, 2018).

[8] A. Bruix, J. T. Margraf, M. Andersen, and K. Reuter, *First-principles-based multiscale modelling of heterogeneous catalysis*, Nat. Catal. **2**, 659 (2019).

[9] L. Foppa, M. Iannuzzi, C. Copéret, and A. Comas-Vives, *Adlayer Dynamics Drives CO Activation in Ru-Catalyzed Fischer–Tropsch Synthesis*, ACS Catalysis **8**, 6983 (2018).

[10] C. M. Friend and B. Xu, *Heterogeneous Catalysis: A Central Science for a Sustainable Future*, Acc. Chem. Res. **50**, 517 (2017).

[11] L. Chanussot, A. Das, S. Goyal, T. Lavril, M. Shuaibi, M. Riviere, K. Tran, J. Heras-Domingo, C. Ho, W. Hu, A. Palizhati, A. Sriram, B. Wood, J. Yoon, D. Parikh, C. L. Zitnick, and Z. Ulissi, *Open Catalyst 2020 (OC20) Dataset and Community Challenges*, ACS Catal. **11**, 6059 (2021).

[12] Z. W. Ulissi, A. Medford, T. Bligaard, and J. K. Nørskov, *To address surface reaction network complexity using scaling relations machine learning and DFT calculations*, Nat. Commun. **8**, 14621 (2017).

[13] J. T. Margraf and K. Reuter, *Systematic Enumeration of Elementary Reaction Steps in Surface Catalysis*, ACS Omega **4**, 3370 (2019).

[14] M. Born and R. Oppenheimer, *Zur Quantentheorie der Molekeln*, Ann Phys. **389**, 457 (1927).

[15] W. Koch and M. Holthausen, *A Chemist's Guide to Density Functional Theory*, 2nd ed. (Wiley-VCH, 2009).

[16] P. Hohenberg and W. Kohn, *Inhomogeneous Electron Gas*, Phys. Rev. **136**, B864 (1964).

[17] W. Kohn and L. J. Sham, *Self-Consistent Equations Including Exchange and Correlation Effects*, Phys. Rev. **140**, A1133 (1965).

[18] D. Marx and J. Hutter, *Ab Initio Molecular Dynamics: Basic Theory and Advanced Methods* (Cambridge University Press, Cambridge, England, 2009).

[19] A. K. Rappe, C. J. Casewit, K. S. Colwell, W. A. Goddard, and W. M. Skiff, *UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations*, J. Am. Chem. Soc. **114**, 10024 (1992).

[20] K. Vanommeslaeghe, E. Hatcher, C. Acharya, S. Kundu, S. Zhong, J. Shim, E. Darian, O. Guvench, P. Lopes, I. Vorobyov, and A. D. Mackerell Jr., *CHARMM general force field: A force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields*, J. Comput. Chem. **31**, 671 (2010).

[21] T. P. Senftle, S. Hong, M. M. Islam, S. B. Kylasa, Y. Zheng, Y. K. Shin, C. Junkermeier, R. Engel-Herbert, M. J. Janik, H. M. Aktulga, T. Verstraelen, A. Grama, and A. C. T. van Duin, *The ReaxFF reactive force-field: development, applications and future directions*, Npj Comput. Mater. **2**, 15011 (2016).

[22] F. Vitalini, A. S. J. S. Mey, F. Noé, and B. G. Keller, *Dynamic properties of force fields*, J. Chem. Phys. **142**, 084101 (2015).

[23] K. Lindorff-Larsen, P. Maragakis, S. Piana, M. P. Eastwood, R. O. Dror, and D. E. Shaw, *Systematic Validation of Protein Force Fields against Experimental Data*, PLOS ONE **7**, 1 (2012).

[24] J. Hoja and A. Tkatchenko, *First-principles stability ranking of molecular crystal polymorphs with the DFT+MBD approach*, Faraday Discuss. **211**, 253 (2018).

[25] O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schütt, A. Tkatchenko, and K.-R. Müller, *Machine Learning Force Fields*, Chem. Rev. **121**, 10142 (2021).

[26] J. A. Keith, V. Vassilev-Galindo, B. Cheng, S. Chmiela, M. Gastegger, K.-R. Müller, and A. Tkatchenko, *Combining Machine Learning and Computational Chemistry for Predictive Insights Into Chemical Systems*, Chem. Rev. **121**, 9816 (2021).

[27] V. L. Deringer, A. P. Bartók, N. Bernstein, D. M. Wilkins, M. Ceriotti, and G. Csányi, *Gaussian Process Regression for Materials and Molecules*, Chem. Rev. **121**, 10073 (2021).

[28] J. Behler, *Constructing high-dimensional neural network potentials: A tutorial review*, Int. J. Quantum Chem. **115**, 1032 (2015).

[29] A. P. Bartók and G. Csányi, *Gaussian approximation potentials: A brief tutorial introduction*, Int. J. Quantum Chem. **115**, 1051 (2015).

[30] M. Rupp, *Machine learning for quantum mechanics in a nutshell*, Int. J. Quantum Chem. **115**, 1058 (2015).

[31] V. L. Deringer, M. A. Caro, and G. Csányi, *Machine Learning Interatomic Potentials as Emerging Tools for Materials Science*, Adv. Mater. **31**, 1902765 (2019).

[32] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, *Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons*, Phys. Rev. Lett. **104**, 136403 (2010).

[33] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller, *Machine learning of accurate energy-conserving molecular force fields*, Sci. Adv. **3**, e1603015 (2017).

[34] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, *Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning*, Phys. Rev. Lett. **108**, 058301 (2012).

[35] A. P. Bartók, S. De, C. Poelking, N. Bernstein, J. R. Kermode, G. Csányi, and M. Ceriotti, *Machine learning unifies the modeling of materials and molecules*, Sci. Adv. **3**, e1701816 (2017).

[36] J. Behler and M. Parrinello, *Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces*, Phys. Rev. Lett. **98**, 146401 (2007).

[37] K. Schütt, P.-J. Kindermans, H. E. Sauceda Felix, S. Chmiela, A. Tkatchenko, and K.-R. Müller, *SchNet: A continuous-filter convolutional neural network for modeling quantum interactions*, in Adv. Neural Inf. Process. Syst. (2017).

[38] O. T. Unke and M. Meuwly, *PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges*, J. Chem. Theory Comput. **15**, 3678 (2019).

[39] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, *E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials*, Nat. Commun. **13**, 2453 (2022).

[40] J. Gasteiger, J. Groß, and S. Günnemann, *Directional Message Passing for Molecular Graphs*, in ICLR (2020).

[41] J. Gasteiger, F. Becker, and S. Günnemann, *GemNet: Universal Directional Graph Neural Networks for Molecules*, in NeurIPS (2021).

[42] O. T. Unke, S. Chmiela, M. Gastegger, K. T. Schütt, H. E. Sauceda, and K.-R. Müller, *SpookyNet: Learning force fields with electronic degrees of freedom and nonlocal effects*, Nat. Commun. **12**, 7273 (2021).

[43] A. V. Shapeev, *Moment Tensor Potentials: A Class of Systematically Improvable Interatomic Potentials*, Multiscale Model Simul. **14**, 1153 (2016).

[44] G. Montavon, M. Rupp, V. Gobre, A. Vazquez-Mayagoitia, K. Hansen, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, *Machine learning of molecular electronic properties in chemical compound space*, New J. Phys. **15**, 095003 (2013).

[45] J. S. Smith, O. Isayev, and A. E. Roitberg, *ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost*, Chem. Sci. **8**, 3192 (2017).

[46] L. Zhang, J. Han, H. Wang, R. Car, and W. E, *Deep Potential Molecular Dynamics: A Scalable Model with the Accuracy of Quantum Mechanics*, Phys. Rev. Lett. **120**, 143001 (2018).

[47] F. A. Faber, A. S. Christensen, B. Huang, and O. A. von Lilienfeld, *Alchemical and structural distribution based representation for universal quantum machine learning*, J. Chem. Phys **148**, 241717 (2018).

[48] B. Huang and O. A. von Lilienfeld, *Communication: Understanding molecular representations in machine learning: The role of uniqueness and target similarity*, J. Chem. Phys. **145**, 161102 (2016).

[49] C. R. Collins, G. J. Gordon, O. A. von Lilienfeld, and D. J. Yaron, *Constant size descriptors for accurate machine learning models of molecular properties*, J. Chem. Phys. **148**, 241718 (2018).

[50] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, *SchNet – A deep learning architecture for molecules and materials*, J. Chem. Phys. **148**, 241722 (2018).

[51] G. Pilania, J. Gubernatis, and T. Lookman, *Multi-fidelity machine learning models for accurate bandgap predictions of solids*, Comput. Mater. Sci. **129**, 156 (2017).

[52] M. Todorović, M. U. Gutmann, J. Corander, and P. Rinke, *Bayesian inference of atomistic structure in functional materials*, Npj Comput. Mater. **5**, 35 (2019).

[53] W. Rawat and Z. Wang, *Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review*, Neural Comput. **29**, 2352 (2017).

[54] Y. Bengio, R. Ducharme, and P. Vincent, *A Neural Probabilistic Language Model*, in Adv. Neural Inf. Process. Syst. (2000).

[55] L. Ruddigkeit, R. van Deursen, L. C. Blum, and J.-L. Reymond, *Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17*, J. Chem. Inf. Model. **52**, 2864 (2012).

[56] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld, *Quantum chemistry structures and properties of 134 kilo molecules*, Sci. Data **1**, 140022 (2014).

[57] J. Hoja, L. Medrano Sandonas, B. G. Ernst, A. Vazquez-Mayagoitia, R. A. DiStasio Jr., and A. Tkatchenko, *QM7-X, a comprehensive dataset of quantum-mechanical properties spanning the chemical space of small organic molecules*, Sci. Data **8**, 43 (2021).

[58] J.-L. Reymond, R. van Deursen, L. C. Blum, and L. Ruddigkeit, *Chemical space as a source for new drugs*, Med. Chem. Commun. **1**, 30 (2010).

[59] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, *Neural Message Passing for Quantum Chemistry*, in ICML (2017).

[60] F. Jensen, *Introduction to Computational Chemistry*, 2nd ed. (John Wiley & Sons, 2007).

[61] A. Groß, *Theoretical Surface Science*, 2nd ed. (Springer, Berlin, Heidelberg, 2009).

[62] A. Szabo and N. Ostlund, *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory* (Dover Books on Chemistry, 1989).

[63] C. J. Cramer, *Essentials of Computational Chemistry*, 2nd ed. (John Wiley & Sons, 2004).

[64] C. Adamo and V. Barone, *Toward reliable density functional methods without adjustable parameters: The PBE0 model*, J. Chem. Phys. **110**, 6158 (1999).

[65] J. P. Perdew, M. Ernzerhof, and K. Burke, *Rationale for mixing exact exchange with density functional approximations*, J. Chem. Phys **105**, 9982 (1996).

[66] J. P. Perdew, K. Burke, and M. Ernzerhof, *Generalized Gradient Approximation Made Simple*, Phys. Rev. Lett. **77**, 3865 (1996).

[67] S. Habershon, D. E. Manolopoulos, T. E. Markland, and T. F. Miller, *Ring-Polymer Molecular Dynamics: Quantum Effects in Chemical Dynamics from Classical Trajectories in an Extended Phase Space*, Annu. Rev. Phys. Chem. **64**, 387 (2013).

[68] T. Schneider and E. Stoll, *Molecular-dynamics study of a three-dimensional one-component model for distortive phase transitions*, Phys. Rev. B **17**, 1302 (1978).

[69] S. G. Moustafa, A. J. Schultz, and D. A. Kofke, *Effects of thermostatting in molecular dynamics on anharmonic properties of crystals: Application to fcc Al at high pressure and temperature*, J. Chem. Phys **149**, 124109 (2018).

[70] M. Allen and D. Tildesley, *Computer Simulation of Liquids* (Akademic Press, 1987).

[71] B. Burger, P. M. Maffettone, V. V. Gusev, C. M. Aitchison, Y. Bai, X. Wang, X. Li, B. M. Alston, B. Li, R. Clowes, N. Rankin, B. Harris, R. S. Sprick, and A. I. Cooper, *A mobile robotic chemist*, Nature **583**, 237 (2020).

[72] H. Ghoddusi, G. G. Creamer, and N. Rafizadeh, *Machine learning in energy economics and finance: A review*, Energy Econ. **81**, 709 (2019).

[73] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, *An Introduction to Deep Reinforcement Learning*, Found. Trends Mach. Learn. **11**, 219 (2018).

[74] F. Musil, A. Grisafi, A. P. Bartók, C. Ortner, G. Csányi, and M. Ceriotti, *Physics-Inspired Structural Representations for Molecules and Materials*, Chem. Rev. **121**, 9759 (2021).

[75] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller, *Machine learning of accurate energy-conserving molecular force fields*, Sci. Adv. **3**, e1603015 (2017).

[76] A. Stuke, C. Kunkel, D. Golze, M. Todorović, J. T. Margraf, K. Reuter, P. Rinke, and H. Oberhofer, *Atomic structures and orbital energies of 61,489 crystal-forming organic molecules*, Sci. Data **7**, 58 (2020).

[77] S. Wengert, G. Csányi, K. Reuter, and J. T. Margraf, *Data-efficient machine learning for molecular crystal structure prediction*, Chem. Sci. **12**, 4536 (2021).

[78] J. Timmermann, Y. Lee, C. G. Staacke, J. T. Margraf, C. Scheurer, and K. Reuter, *Data-efficient iterative training of Gaussian approximation potentials: Application to surface structure determination of rutile IrO2 and RuO2*, J. Chem. Phys **155**, 244107 (2021).

[79] H. Huo and M. Rupp, *Unified Representation of Molecules and Crystals for Machine Learning*, arXiv:1704.06439, preprint, 2017.

[80] K. Hansen, F. Biegler, R. Ramakrishnan, W. Pronobis, O. A. von Lilienfeld, K.-R. Müller, and A. Tkatchenko, *Machine Learning Predictions of Molecular Properties: Accurate Many-Body Potentials and Nonlocality in Chemical Space*, J. Phys. Chem. Lett. **6**, 2326 (2015).

[81] A. P. Bartók, R. Kondor, and G. Csányi, *On representing chemical environments*, Phys. Rev. B **87**, 184115 (2013).

[82] J. Behler and G. Csányi, *Machine learning potentials for extended systems: a perspective*, Eur. Phys. J. B **94**, 142 (2021).

[83] C. G. Staacke, S. Wengert, C. Kunkel, G. Csányi, K. Reuter, and J. T. Margraf, *Kernel charge equilibration: efficient and accurate prediction of molecular dipole moments with a machine-learning enhanced electron density model*, Mach. Learn.: Sci. Technol. **3**, 015032 (2022).

[84] C. G. Staacke, H. H. Heenen, C. Scheurer, G. Csányi, K. Reuter, and J. T. Margraf, *On the Role of Long-Range Electrostatics in Machine-Learned Interatomic Potentials for Complex Battery Materials*, ACS Appl. Energy Mater. **4**, 12562 (2021).

[85] J. Gasteiger, S. Giri, J. T. Margraf, and S. Günnemann, *Fast and Uncertainty-Aware Directional Message Passing for Non-Equilibrium Molecules*, in ML for Molecules workshop, NeurIPS (2020).

[86] K. Schütt, O. Unke, and M. Gastegger, *Equivariant message passing for the prediction of tensorial properties and molecular spectra*, in ICML (2021).

[87] V. L. Deringer and G. Csányi, *Machine learning based interatomic potential for amorphous carbon*, Phys. Rev. B **95**, 094203 (2017).

[88] B. Cheng, R.-R. Griffiths, S. Wengert, C. Kunkel, T. Stenczel, B. Zhu, V. L. Deringer, N. Bernstein, J. T. Margraf, K. Reuter, and G. Csanyi, *Mapping Materials and Molecules*, Acc. Chem. Res. **53**, 1981 (2020).

[89] S. De, A. P. Bartók, G. Csányi, and M. Ceriotti, *Comparing molecules and solids across structural and alchemical space*, Phys. Chem. Chem. Phys. **18**, 13754 (2016).

[90] M. F. Langer, A. Goeßmann, and M. Rupp, *Representations of molecules and materials for interpolation of quantum-mechanical simulations via machine learning*, Npj Comput. Mater. **8**, 41 (2022).

[91] L. Himanen, M. O. J. Jäger, E. V. Morooka, F. Federici Canova, Y. S. Ranawat, D. Z. Gao, P. Rinke, and A. S. Foster, *DScribe: Library of descriptors for machine learning in materials science*, Comput. Phys. Commun. **247**, 106949 (2020).

[92] A. N. Tikhonov, A. Goncharsky, V. V. Stepanov, and A. G. Yagola, *Numerical Methods for the Solution of Ill-Posed Problems* (Kluwer Academic, Dordrecht, 1995).

[93] R. Caruana, S. Lawrence, and C. Giles, *Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping*, in Adv. Neural Inf. Process. Syst. (2000).

[94] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*, J. Mach. Learn. Res. **15**, 1929 (2014).

[95] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016).

[96] J. Kukačka, V. Golkov, and D. Cremers, *Regularization for Deep Learning: A Taxonomy*, arXiv:1710.10686, preprint, 2017.

[97] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods* (Cambridge Univ. Press, 2000).

[98] G. Csányi, M. J. Willatt, and M. Ceriotti, Machine-Learning of Atomic-Scale Properties Based on Physical Principles, in *Machine Learning Meets Quantum Physics* (Springer International Publishing, Cham, 2020).

[99] B. Schölkopf, R. Herbrich, and A. J. Smola, *A Generalized Representer Theorem*, in Computational learning theory (2001).

[100] J. Quiñonero-Candela and C. E. Rasmussen, *A Unifying View of Sparse Approximate Gaussian Process Regression*, J. Mach. Learn. Res. **6**, 1939 (2005).

[101] I. Jolliffe, *Principal Component Analysis*, 2nd ed. (Springer, New York, NY, 2002).

[102] B. Schölkopf, A. Smola, and K.-R. Müller, *Nonlinear Component Analysis as a Kernel Eigenvalue Problem*, Neural Comput. **10**, 1299 (1998).

[103] T. Mueller, A. Hernandez, and C. Wang, *Machine learning for interatomic potential models*, J. Chem. Phys **152**, 050902 (2020).

[104] A. Musaelian, S. Batzner, A. Johansson, L. Sun, C. J. Owen, M. Kornbluth, and B. Kozinsky, *Learning Local Equivariant Representations for Large-Scale Atomistic Dynamics*, arXiv:2204.05249, preprint, 2022.

[105] J. Snoek, H. Larochelle, and R. P. Adams, *Practical Bayesian Optimization of Machine Learning Algorithms*, in Adv. Neural Inf. Process. Syst. (2012).

[106] S. Günnemann, *Lecture notes in Machine Learning*, Nov. 2018.

[107] S. Wengert, *mltools*, `https://github.com/simonwengert/mltools.git`.

[108] A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. B. Jensen, J. Kermode, J. R. Kitchin, E. L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, *The atomic simulation environment—a Python library for working with atoms*, J. Phys.: Condens. Matter **29**, 273002 (2017).

Appendices

Paper # 1

Machine Learning in Chemical Reaction Space
Sina Stocker, Gábor Csányi, Karsten Reuter and Johannes T. Margraf
Nat. Commun. 11, 5505 (2020).
DOI: https://doi.org/10.1038/s41467-020-19267-x

nature COMMUNICATIONS

ARTICLE

https://doi.org/10.1038/s41467-020-19267-x OPEN

Machine learning in chemical reaction space

Sina Stocker [1], Gábor Csányi [2], Karsten Reuter [1,3] & Johannes T. Margraf [1✉]

Chemical compound space refers to the vast set of all possible chemical compounds, estimated to contain 10^{60} molecules. While intractable as a whole, modern machine learning (ML) is increasingly capable of accurately predicting molecular properties in important subsets. Here, we therefore engage in the ML-driven study of even larger reaction space. Central to chemistry as a science of transformations, this space contains all possible chemical reactions. As an important basis for 'reactive' ML, we establish a first-principles database (Rad-6) containing closed and open-shell organic molecules, along with an associated database of chemical reaction energies (Rad-6-RE). We show that the special topology of reaction spaces, with central hub molecules involved in multiple reactions, requires a modification of existing compound space ML-concepts. Showcased by the application to methane combustion, we demonstrate that the learned reaction energies offer a non-empirical route to rationally extract reduced reaction networks for detailed microkinetic analyses.

[1] Chair of Theoretical Chemistry and Catalysis Research Center, Technische Universität München, Garching, Germany. [2] Engineering Laboratory, University of Cambridge, Cambridge CB2 1PZ, UK. [3] Fritz-Haber-Institut der Max-Planck-Gesellschaft, Berlin, Germany. ✉email: johannes.margraf@ch.tum.de

Reaction networks are essential tools for the description, illustration, and fundamental understanding of chemical processes in such diverse fields as catalysis[1–4], combustion[5–7], polymerization[8], atmospheric chemistry[9], systems chemistry[10,11], and the origin of life[12]. Indeed, any study of chemical kinetics or selectivity is essentially a study of a reaction network. In many cases, however, the understanding of complex chemical processes is hampered by the sheer size of the networks in question[1,13–21]. For example, we recently reported a database of over 1 million elementary reactions for molecules no larger than four non-hydrogen atoms containing carbon, oxygen and hydrogen[22].

The reaction networks typically used in microkinetic studies of natural and industrial processes are therefore necessarily merely sub-graphs of the full network of possible reactions (see Fig. 1)[20,23]. This is not automatically a problem, as large parts of the latter may not be thermodynamically accessible. It is therefore entirely possible that a microkinetic model based on a reduced reaction network correctly describes the overall kinetics of a complex process[1,6,20]. Meanwhile, the big advantage of focusing on sub-graphs is that the kinetics and thermochemistry of each elementary step may be explicitly computed from first principles. This offers a non-empirical route to understanding complex reaction mechanisms.

Notwithstanding, the difficulty lies in knowing which parts of the full network to keep. One would need at least an approximate notion of the reaction thermochemistry (and ideally the kinetics) of the full network, to be able to do this on a rational basis. This information is typically not available. Indeed, not even the topology of the full network is usually taken into account. Instead, state-of-the-art reaction networks are generally built by hand, based on chemical intuition and (sparse) experimental evidence. The frequently observed failure to correctly predict the selectivities of complex catalytic processes with first–principles microkinetics indicates that such ad hoc networks may miss important links[24–26].

The central impediment towards a non-empirical construction of reduced reaction networks is the large computational cost of first-principles electronic structure methods such as density-functional theory (DFT). It is simply not feasible to routinely compute tens or hundreds of thousands of reaction energies (REs) and activation barriers. In this context, machine-learning (ML) models that are trained on a limited number of DFT calculations have recently emerged as powerful tools for the high-throughput prediction of molecular and materials properties[27–33]. Simply put, ML can be used to interpolate properties (such as energies) across chemical compound space. State-of-the-art methods actually surpass chemical accuracy (ca. 0.05 eV) when applied to standard benchmarks like the QM9 database[34–38]. Similarly, ML models can be applied to conformational space (e.g., when trained on ab initio molecular dynamics trajectories) or even interpolate across chemical and conformational space at the same time[39–41].

While exploring compound space is useful in its own right (e.g., for drug or materials design), chemistry is the science of transformations in chemical space. In contrast, virtually all ML models for organic molecules to date are trained on reference data derived from the chemical universe database of Reymond and coworkers, which enumerates potentially stable, drug-like molecules[41–43]. Almost by construction, these models therefore cannot describe elementary reactions such as the ones shown in Fig. 1, which typically involve radical or charged intermediates. In our view, the application of ML to areas like catalysis and combustion requires a shift of focus from stable molecules to radicals (i.e., the nodes in Fig. 1) and to reactions (the edges). The goal of this paper is therefore to begin the development of ML models for the exploration of reaction space, as opposed to compound space.

Specifically, we introduce a new DFT database of closed- and open-shell molecules that covers an extensive network of chemical reactions. We then develop ML models to predict atomization and REs. Finally, the models are used to explore the reaction network of methane combustion and identify the most relevant reaction steps and fragments out of a large initial database.

Results

Data and kernels. To train reactive ML models, a reference database of both open and closed-shell systems must be established. A large set of such structures was enumerated using a graph-based approach[22], and the ground-state geometry and energy of each system was determined with DFT calculations using the hybrid PBE0 functional with Tkatchenko-Scheffler dispersion corrections[44–46]. The resulting Rad-6 reference database comprises 10,712 molecules containing carbon, oxygen and hydrogen, the largest of which consists of six non-hydrogen atoms. As illustrated in Fig. 2, this dataset is rich in

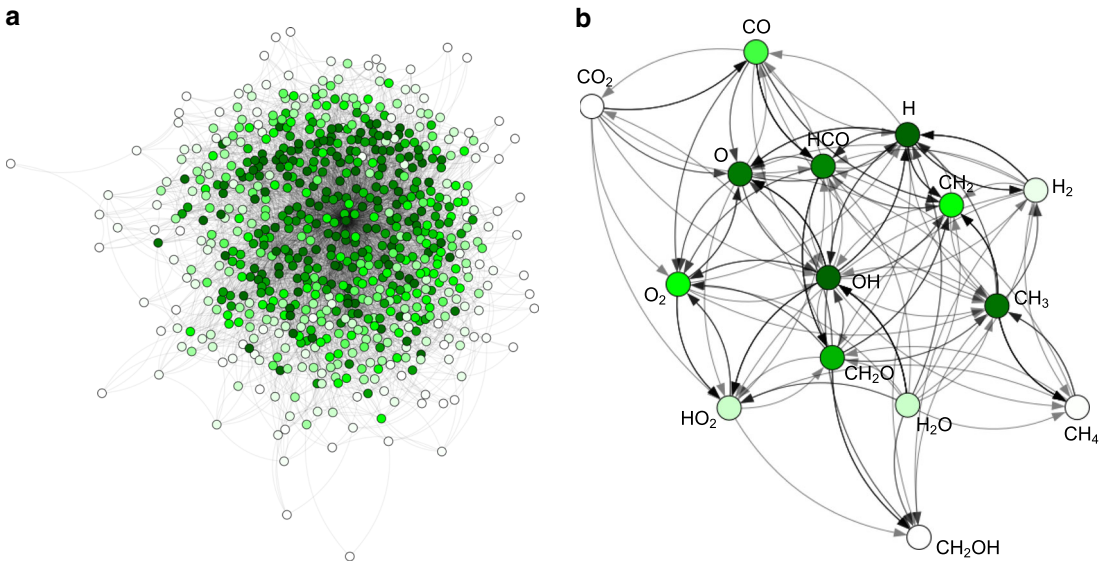


Fig. 1 Visualization of chemical reaction spaces as graphs with molecules as nodes and reactions as edges. a Full network of bond dissociation reactions for carbon-, oxygen-, hydrogen-containing molecules with up to four heavy atoms. **b** Reduced reaction network of the initial steps of natural gas combustion. Nodes are colored according to the number of incident edges/reactions (their degree) from low (white) to high (dark green).

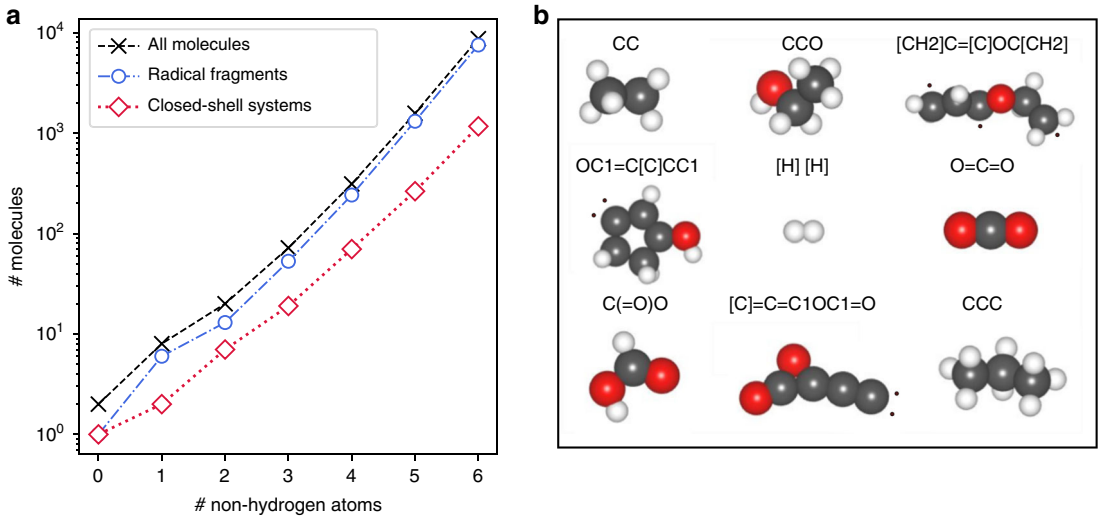


Fig. 2 The Rad-6 database. a Number of molecules in the database, according to their number of non-hydrogen atoms. **b** Structures of representative molecules in the database. Dots indicate radicals and respective SMILES strings are listed.

unconventional structural motifs, such as poly-radicals. As is commonly observed, the space of possible compounds scales exponentially with the system size (see Fig. 2, left). This figure also reveals that radical fragments in fact dominate the database, as they are combinatorically much more frequent (by an order of magnitude) than closed-shell systems. Notably, this dominance of open-shell systems prevails, although more than half of the originally enumerated radicals decomposed or rearranged upon geometry optimization. Importantly, these unstable cases were not included in the database. This choice was made because the definition of a chemical reaction requires the specification of the molecular topologies of educts and products (and how they are transformed). The full Rad-6 database is provided in the supporting information to this article.

Two central quantities that are needed to fully understand the overall kinetics of a reaction network are the RE (E_{reac}, RE) and the activation energy (barrier) for each reaction. Indeed, REs provide the most important features of the reaction network and can in some cases even be used to predict activation energies via the Brønsted-Evans-Polanyi relation[47–49]. Furthermore, while the activation energy is a property of each individual reaction (the edges in a graph), the RE can be computed from molecular atomization energies (E_{at}, AE), i.e. information from pairs of nodes in a graph, meaning that much fewer calculations are required to predict the REs in a large reaction network. Specifically, to predict 1000 REs for 20 molecules, one only needs 20 ground-state geometries. In contrast, predicting the corresponding activation energies would require 1000 additional transition state (TS) geometries. Not only are there more TS geometries, but these are also much harder to obtain, both in terms of computational effort and in terms of the human intervention needed for successful transition state searches. This makes predicting REs the logical first step in the ML-driven exploration of reaction networks.

Specifically, for a reaction of the type:

$$A \longrightarrow B + C, \tag{1}$$

the REs can be computed from molecular atomization energies via:

$$E_{\mathrm{reac}} = E_{\mathrm{at}}^{B} + E_{\mathrm{at}}^{C} - E_{\mathrm{at}}^{A}, \tag{2}$$

where we define the AE without loss of generality as the total energy of the molecule minus total energies of the isolated neutral atoms.

Learning atomization energies across chemical compound space is a well-established practice in the ML literature. In a first approach, we can therefore apply such compound space models for predicting REs, as long as they are trained on a reactive database like Rad-6. Herein, we use Kernel Ridge Regression (KRR) with the SOAP[50] representation, as a state-of-the-art ML method (see SI for details). In brief, KRR uses a kernel function $k(x_i, x_j)$, to measure the similarity between representations x_i and x_j. The herein used SOAP representation is one of a class of atom-density projections that have been found to yield highly accurate molecular ML models[51,52]. With this type of model, the AE of an unknown molecule can be predicted according to its similarity with known molecules in a training set. Since the AE is a molecular property and SOAP is an atomic representation, an additional step is required for evaluating the similarity of molecules. This can, for example, be achieved with the average kernel[37]:

$$K_{\mathrm{int}}(A,B) = \sum_{a\in A, b\in B} \frac{1}{N_A\, N_B} k(x_a, x_b), \tag{3}$$

where N_A and N_B are the numbers of atoms a and b in molecules A and B, respectively, and x_a is the SOAP representation of the chemical environment of atom a. The lower-case k is used to differentiate the atomic from the molecular kernel function K. Alternatively, one can also use the sum kernel:

$$K_{\mathrm{ext}}(A,B) = \sum_{a\in A, b\in B} k(x_a, x_b). \tag{4}$$

Both average and sum kernels have been successfully used in ML models of the AE, but there is a crucial difference in their properties[34,36]: Specifically, the average kernel disregards size differences between molecules. It provides a measure for how similar the atoms in molecule A are to the ones in molecule B, on average. Meanwhile, the non-normalized sum kernel is sensitive to size differences. Consequently, models using the average kernel should be used to predict intensive quantities, and models using the sum kernel should predict extensive properties[53]. Herein, all models using the average kernel are therefore trained on the atomization energy per atom (AE/N, an intensive quantity). The predicted AE/N is afterwards simply multiplied with the number of atoms N to recover the AE. Meanwhile, the sum kernel can directly be trained on (and predict) the AE[53]. In the following we will refer to Eq. (3) as the intensive kernel (K_{int}) and to Eq. (4) as

the extensive kernel (K_{ext}). As an aside, it should be noted that using such linear combination kernels is equivalent to the partitioning of the total energy inherent, for instance, to Gaussian Approximation Potentials[29,36].

To train ML models, the Rad-6 database is split into training, validation (for hyperparameter optimization) and test sets. To obtain representative training sets, we use the farthest point sampling (FPS) method[36]. In FPS, data-points are sequentially selected to maximize the distance between a new data-point (a molecule A) and all previously selected points (molecules B already in the training set). In the present context, this means new molecules added to the training set should be as dissimilar as possible to all previously selected molecules. The distance between molecules is measured using the previously introduced kernels, according to:

$$D(A,B) = \sqrt{K(A,A) + K(B,B) - 2K(A,B)}. \quad (5)$$

Because $D(A, B)$ depends on the kernel, we obtain different training sets for the intensive and extensive kernels. Most importantly, while we normalize K_{int} so that $K_{int}(A, A) = K_{int}(B, B) = 1$, K_{ext} is not normalized. Consequently, $K_{ext}(A, A) \sim N_A^2$ and $K_{ext}(B, B) \sim N_B^2$. This means that the distance $D_{ext}(A, B)$ evaluated with the extensive kernels tends to be greater between large systems than the distance between small systems. Accordingly, mostly large molecules are selected during the early iterations of FPS with D_{ext}, whereas the intensive distance D_{int} maximizes the average chemical diversity in the training set irrespective of size. It should be noted that a FPS selection based on maximally diverse atomic environments rather than molecules (e.g. using a softmax criterion[54]) would also be possible. This may be a better choice for datasets with large molecules.

Beyond their use in regression methods like KRR, kernels can also be used for dimensionality reduction and visualization of large data sets with the kernel principal component analysis (kPCA) method[55,56]. In Fig. 3, kPCA plots of the Rad-6 chemical compound space for the intensive and extensive kernels are shown. Here, the two principal components mainly reflect the degree of saturation (the number of hydrogen atoms) and the oxygen/carbon ratio. The main difference in both projections is that the extensive kernel additionally displays a size-dependence, with small molecules (up to 4 heavy atoms) concentrated in the bottom right corner (see SI for more details).

Superposed on the projected landscapes, Fig. 3 shows the color-coded variation of the DFT computed AEs. A clear trend from more negative values in the top right to less negative values in the bottom left can be discerned for K_{int}. This correlation of AE/N with the degree of saturation results simply because highly saturated molecules contain only single bonds, while unsaturated molecules contain double and triple bonds. The gradual variation of both AE and AE/N also provides an intuitive understanding of why kernel models work for predicting molecular energies: Molecules that are close in the kPCA plot (i.e., considered to be similar by the kernel) also have a similar AE. Finally, Fig. 3 also illustrates the distribution of the FPS-selected training points, which evenly cover the compound space, but also span most of the more isolated points at the bottom of the figure.

Machine learning in compound space. In Fig. 4, the learning curves for AE predictions with the extensive and intensive kernels and using both D_{ext}-based and D_{int}-based FPS sets are shown, i.e., we also combine extensive kernel learning with intensive training sets and vice versa. It can be seen that with the largest training sets, all four models are able to predict atomization energies for these systems with mean absolute errors (MAEs) well below 0.1 eV. In all cases, the log-log plots display the expected linear relationship (i.e., the learning curve can be fitted as a power law), indicating that even higher accuracy could be achieved with more data. To put this performance into perspective, it should be noted that our baseline method (dispersion-corrected hybrid DFT) itself has an average accuracy of ca. 4–5 kcal mol^{-1} (0.2 eV) for REs and barriers[57,58].

Additional ML models were trained on randomly sampled training sets, to provide a baseline for the FPS schemes. The corresponding AE learning curves are comparable to the extensive FPS (see SI). As has previously been noted, random sampling is actually advantageous for very small training sets, but the learning rate is lower than for both FPS schemes translating into inferior performance for larger training sets[36].

Following common practice, all errors are shown for the total AE, even for the intensive models. Clearly, this is not a completely fair comparison, as the intensive models are trained to minimize the AE/N and not the total AE error. This explains

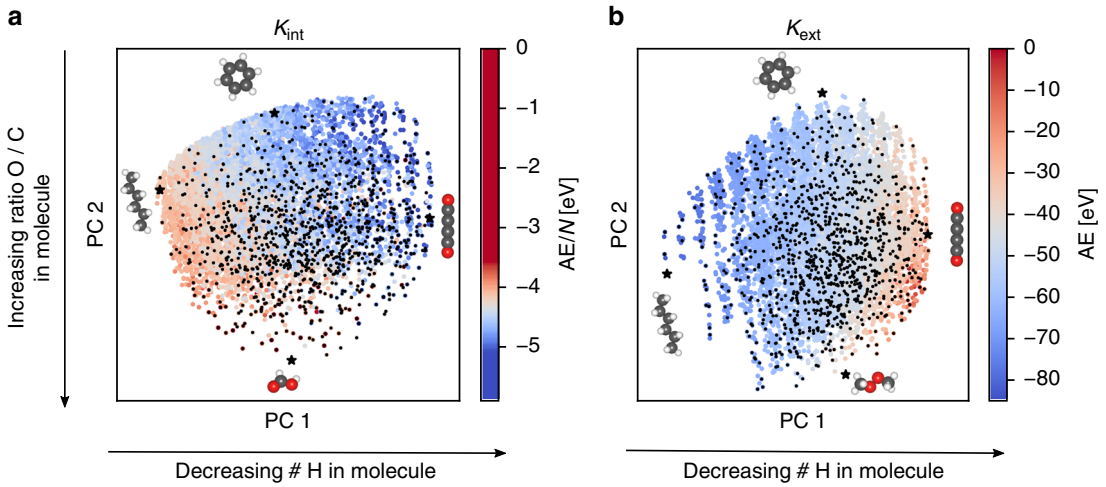


Fig. 3 Visualizing Rad-6 with Kernel Principal Component Analysis (kPCA). a kPCA based on an intensive kernel. **b** kPCA based on an extensive kernel. Points are colored according to the DFT atomization energy per atom in (**a**) and total atomization energy in (**b**). The arrows provide a qualitative interpretation of the principal component (PC) axes and small black dots indicate the FPS-selected training configurations for a ML model with 1000 training molecules and using the corresponding distance criterion (D_{int} (**a**), D_{ext} (**b**)), see text.

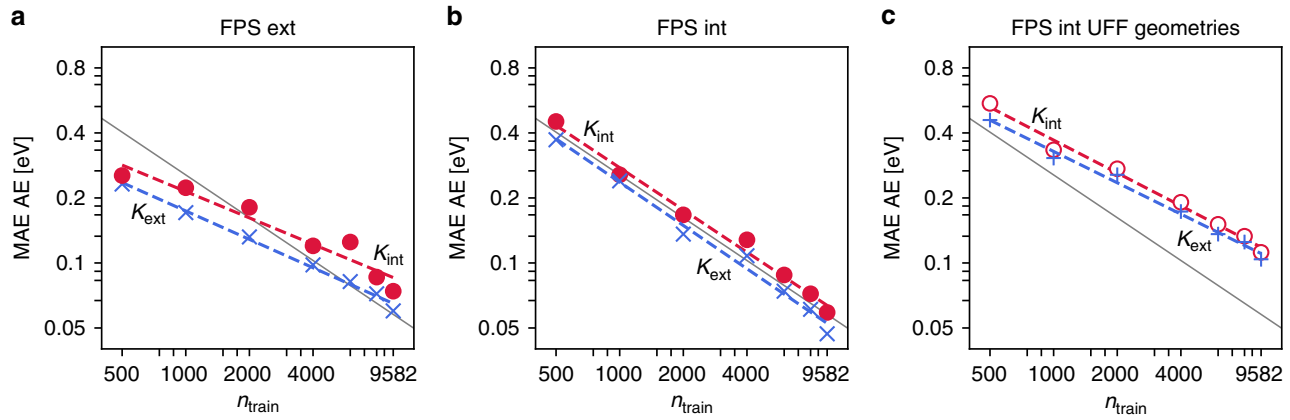


Fig. 4 Learning curves for atomization energies (AE). a Mean absolute error (MAE) of AE predictions on the test set, as a function of the number of training molecules n_{train}. The training sets were constructed using FPS with the extensive (**a**) and intensive kernels (**b**) (see text). **c** AE learning curves using molecular geometries obtained with the universal forcefield (UFF). The gray line represents a learning rate of $n_{train}^{0.65}$ and serves as a guide to the eye in all three panels.

the seemingly counter-intuitive fact that the extensive model performs better even on the intensive FPS training set. It has been suggested in the context of electronic structure methods that AE/N may generally be a more appropriate target for fitting and benchmarking[58,59]. Specifically, fitting on the total AE will selectively favor large systems over small ones, as they offer a larger potential for improvement in the loss function. This also carries over to the FPS selection, as extensive selection will initially focus on larger molecules which are deemed to be more dissimilar than smaller ones. We will see later that this has significant consequences for reaction networks and REs. Nevertheless, based on the data in Fig. 4 one would deduce a slight superiority of the extensive kernel.

Fully optimized DFT geometries will unfortunately not be available for ML training and prediction in a realistic application. If they were, the DFT energy would be known and the ML prediction would be redundant[60]. We therefore also used simple forcefield geometries (based on the universal forcefield, UFF)[61] for training and prediction, still using the ground-state energies of relaxed DFT geometries as the target property. As shown in Fig. 4c and detailed in the SI, all trends discussed for the DFT geometries are unchanged, but the MAEs are somewhat higher, roughly by a factor of two. Such inferior performance of ML models using approximate geometries has also been observed for closed-shell data sets like QM9, but it is more pronounced here[36]. This reflects the fact that general forcefields like UFF are not designed for the description of radicals, which make up a large part of Rad-6. In this context, semi-empirical electronic structure methods might offer an alternative low-cost method for more reliable geometries[62,63]. Note however that such methods will invariably afford some amount of rearrangement and decomposition upon geometry optimization, which would introduce a mismatch between the structure used to build the SOAP representation and the structure for which the target energies are computed. This could in principle be mitigated by using constrained relaxations, but defining universal geometrical constraints in a high-throughput setting is not trivial.

It has also been shown that predictions from approximate geometries can be improved by using a measure of the quality of the training geometries to adjust the model regularization for each training sample[36]. As shown in the SI, this is not successful for Rad-6. Again, we attribute this to the overall poor and inconsistent quality of the UFF geometries for open-shell systems, highlighting another challenge when moving towards ML approaches for reaction space.

Nonetheless, even UFF-based models with fairly small training sets already provide a reasonable estimate of the AEs across chemical compound space. This is illustrated in Fig. 5, where an interpolated AE/N surface for an ML model trained on 1000 UFF structures is compared to the DFT reference values. The plots are visually almost indistinguishable. This serves to emphasize that even a ML model trained on 10% of the database already provides an adequate representation of its overall thermochemistry. Recall that the core task for the development of rationally reduced reaction networks is not an excessive accuracy of this thermochemistry as typically targeted in existing ML work for compound space. Instead, the overall topology needs to be appropriately represented to a degree that enables the selection or dismissal of reactions when building sub-graphs.

Machine learning in reaction space. With the ML-predicted AEs, one can readily calculate REs using Eq. (2), in strict analogy to how they are computed with first-principles methods. In this case, errors in the predicted AEs will propagate to the predicted REs. Under the most basic assumptions (i.e., an uncorrelated, constant uncertainty σ_{AE} for every AE prediction), one would expect the uncertainty in the RE prediction for a reaction $A \longrightarrow B + C$ to be $\sqrt{3}\sigma_{\mathrm{AE}}$. While this is a very rough estimate, it indicates that we would generally expect the error on REs to correlate with the AE error, and that the former should be larger than the latter.

To test these expectations, a reaction network containing 32,515 bond-breaking reactions, Rad-6-RE, was generated using the Rad-6 molecules (see SI for details and the full dataset). In Fig. 6, we show the relation between the performance of different ML models for AE and RE predictions, using both FPS training set selections (multiple points for each method correspond to the different training set sizes shown in Fig. 4). These plots reveal several interesting trends. As expected, the RE error correlates with the AE error. However, there are significant differences both with respect to the FPS selection and the kernels. Most notably, all models display unexpectedly large errors for the smaller ($N \leq$ 2000) extensive training sets. In contrast, the models trained on the intensive FPS display RE errors that are much closer to the corresponding AE errors. Strikingly, the combination of intensive kernel learning and intensive training set selection leads to RE

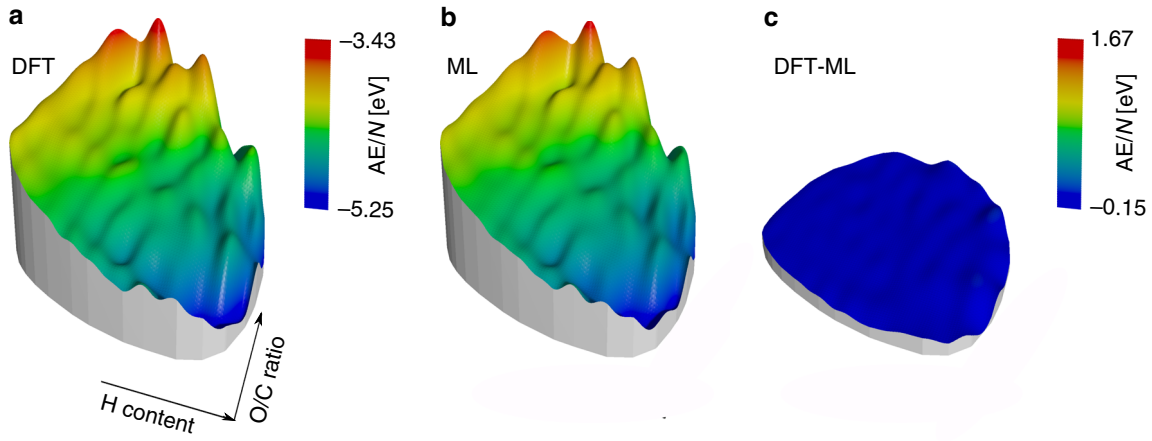


Fig. 5 Illustration of the Rad-6 chemical space as an interpolated height profile. a kPCA as in Fig. 3 showing the DFT reference intensive atomization energies AE/*N* (in eV). **b** Prediction from the ML model using an intensive kernel and a small intensively selected training set of only 1000 molecules with UFF geometries. **c** Respective differences (DFT-ML). Here, the range of the colorbar is shifted but the scale is the same.

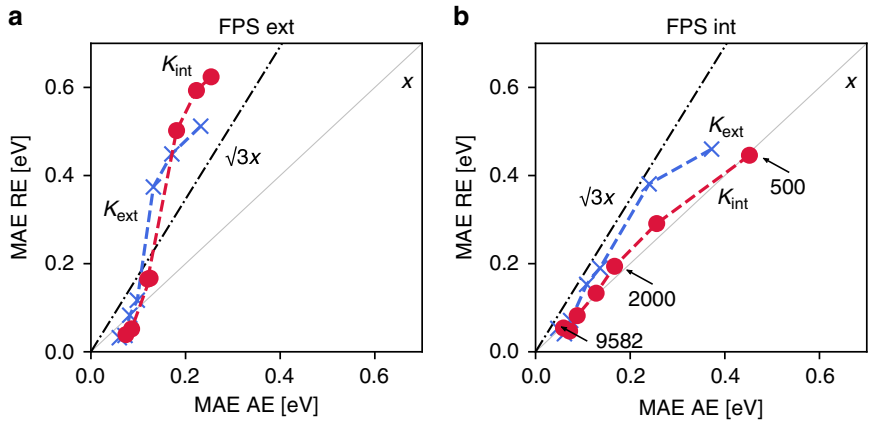


Fig. 6 Correlation of mean absolute errors (MAE) for AE and RE prediction. a Correlation plot for the extensive FPS training set using the extensive and intensive kernels and DFT geometries. **b** Correlation plot for the intensive FPS training set using the extensive and intensive kernels and DFT geometries. Multiple points for each model represent the different training set sizes shown in Fig. 4 (indicated in (**b**)), with smaller AE errors corresponding to larger training sets.

errors that are almost identical to the corresponding AE errors across all training set sizes.

These observations can be understood in light of the fact that not all molecules are equally weighted in a reaction network. As can be seen in Fig. 1, some molecules are central hubs in the network (dark green), whereas others lie on the periphery and only contribute to few reactions (white)[11,19]. The existence of such hubs, which correspond to molecules with dramatically higher importance, is a fundamental difference between reaction space and the homogeneously weighted chemical compound space. In Rad-6-RE, the most important such hubs are small molecules that correspond to functional groups (OH, CH_3, etc.) and the isolated atoms C, H and O. As mentioned previously, the extensive kernel distance D_{ext} will consider all smaller molecules to be more similar in terms of their kernel distance (Eq. (5)), because the terms $K_{ext}(A, A)$ and $K_{ext}(B, B)$ scale with the number of atoms. Small molecules are therefore selected later in an extensive FPS selection, and are consequently absent from the smaller training sets. This can lead to relatively large errors on important hub molecules, which will consequently have an outsized impact on the RE error.

In other words, the large discrepancy between RE and AE for small extensive training sets is because small molecules are less likely to be included. This notion is further reinforced by considering the performance of the models based on random sampling. While the AE predictions of these models are of comparable accuracy with the FPS models (in particular for the smaller training sets), the performance for RE prediction is very poor, with MAEs above 1 eV for small training sets (see SI). Even when the extensive kernel is trained on intensive sets, smaller molecules still offer less potential for improving the loss function and thus lead to a poorer performance for REs.

In complete contrast to the situation in compound space, an intensive kernel with an intensively selected training set is therefore a better choice for ML models in reaction space. This indicates that some of the experience gathered hitherto for ML in chemical compound space (like the significant work on the QM9 database)[34–36] will not necessarily carry over to reaction spaces. Realizing the particular relevance of hub molecules, a straightforward adaptation could for instance simply be to inversely scale the extensive distance used in the FPS selection by the degree of the node in the reaction network, i.e., by the number of reactions in which the molecule is involved (see Fig. 1). Similarly, the least-squares problem for an extensive kernel could be adjusted by weighting the molecules according to their inverse size. With this work, we hope to initiate such dedicated methodological development for reaction spaces and will pursue corresponding research in the future.

It should also be noted that the special topology of reaction networks makes model evaluation for REs in a strict statistical learning framework difficult. The reaction network Rad-6-RE contains most of the Rad-6 molecules. Computing the REs for this network is therefore not a pure prediction, as some molecules in each reaction may be in the training set. In principle, it would be desirable to evaluate the performance on a separate reaction network that contains no training molecules at all. However, this can only be achieved in two ways: Either the test network contains no small molecules like CO and OH, or these molecules are excluded from the training set. The former option leads to a very unnatural reaction network, that misses the most frequent classes of bond-breaking events. Meanwhile, the latter option leads to a very poor training set, and thus an overly pessimistic estimate of model performance.

We therefore decided not to follow this strict separation of training and prediction for the RE MAEs shown in Fig. 6. This also explains why the RE error is in some cases actually lower than the AE error, contrary to expectation: The RE MAE benefits from the fact that the prediction error of all tested models is somewhat lower on the training sets (see SI). Indeed, KRR models can in practice display a negligible error on the training set if the regularization parameter is chosen to be very small, as is advocated by some authors[64].

Exploration of reaction networks. Finally, we return to the original motivation of this work, namely the ML-aided exploration of complex reaction networks. To illustrate the use of ML-predicted REs, we consider a closed network of over 21,393 elementary reactions, containing a large variety of bond-breaking, transfer and rearrangement reactions for oxygen, carbon and hydrogen-containing molecules[22]. Note that this network is deliberately not a subset of Rad-6, although there is significant overlap (ca. 80% of the involved molecules are included in Rad-6). This is thus, at least partially, an out-of-sample application. The challenge lies in determining which of the elementary reactions are likely relevant to a chemical process of interest. As an exemplary process we consider the early stages of methane combustion[65–69].

To validate the proposed ML models for this application, additional DFT calculations were performed on the out-of-sample systems. Unfortunately, these systems mostly decompose or rearrange upon DFT geometry optimization. Note that this does not necessarily mean that they are inherently unstable, however, just that the corresponding local minima were not found when starting from a (inaccurate) UFF geometry. We therefore used DFT single point calculations on UFF geometries here. Overall, a good correlation between DFT and ML-predicted energies is found, with systematically lower ML AEs (see SI). This systematic bias can easily be understood since the ML models predict the DFT energies of relaxed geometries, but the validation energies are for frozen UFF geometries. The latter is by definition larger than the former. This shows that the ML model can be used to estimate relaxed DFT energies even when these are not readily available from DFT calculations.

To qualitatively explore this network, a mean-field microkinetic simulation of the reaction of equal parts CH_4 and O_2 was performed, assuming a constant activation barrier for all reactions (see SI for details). Under these assumptions, the reaction dynamics are only driven by the REs and the law of mass-action. While the true activation energies and detailed reaction conditions (initial concentrations, temperature, pressure, etc.) will obviously play a crucial role for the actual mechanism, such a simplified microkinetic simulation provides insight into how thermochemistry and the topology of the reaction network define which intermediates and reaction steps are at all relevant to the process. By observing how the reaction network grows with simulation time, we can furthermore understand how intermediates and reactions sequentially become available, as mass flows through different paths of the network. Only requiring ML-predicted REs as input, such a simulation is therefore a first step towards the envisioned rational reduction of the full network to tractable sub-graphs.

Figure 7 summarizes the results obtained based on the intensive kernel ML model trained with an intensively selected FPS set of 9582 UFF structures. Shown are the reduced reaction networks extracted as those parts of the full network that are accessed at increasing simulation times. These reduced networks are highly revealing, as they form a hierarchy of different chemistries relevant to combustion. For example, in line with general expectations[6], the smallest network contains peroxide chemistry, with the hydrogen transfer from methane to molecular oxygen as the dominant pathway. Subsequently formed CO_yH_x intermediates also comprise generally anticipated molecules like methanol (CH_3OH) or formic acid (HCOOH), but also more exotic species like the Criegee intermediate (CH_2OO). Interestingly, the formation of the main product CO_2 only appears in larger subgraphs after dimerization reactions have already led to C_2 intermediates like ethylene (C_2H_4) and ethane (C_2H_6). Finally, the largest subgraphs shown include already more complex molecules like propane (C_3H_8) and propene (C_3H_6) and comprise a total of 887 reactions.

It should be emphasized that the networks in Fig. 7 are not intended to represent a definitive mechanism for methane combustion, not least because this mechanism strongly depends on reactions conditions like temperature, pressure and the methane/oxygen ratio[68]. Instead, this analysis provides insight into what intermediates and elementary steps should be considered when constructing reduced reaction networks for mechanistic studies. While assuming constant barriers is clearly a harsh approximation in a microkinetic simulation, we note that predicting activation energies for the full network is not necessary to extract the relevant reduced reaction network for subsequent analysis. In many cases, an elementary reaction can be discarded because of a large thermochemical barrier alone. In other words, if a reaction is found to be irrelevant in a microkinetic simulation with constant barriers, it will not become relevant once activation barriers are included. Of course, activation barriers for the reduced network must still be computed for a quantitative microkinetic simulation, but this is only a small subset of the full network.

Note also that a pure ML approach may miss important domain knowledge. For example, both singlet and triplet spin-states of CH_2 are relevant in combustion[6,68]. Instead, the graph-based enumeration approach[22] used to generate Rad-6, generically only considers the lowest-spin state of each molecule (with manually implemented exceptions of triplet O_2 and the isolated atoms to prevent completely unphysical results). Nonetheless, our pure ML approach finds all intermediates considered in empirical reduced methane combustion mechanism like the skeletal mechanism of Lu et al.[6,70]. On the other hand, the unbiased nature of ML approaches has the benefit of providing unexpected suggestions that would perhaps not be considered otherwise. For example, already our proof-of-concept reduced reaction networks of methane combustion suggest a pathway for CO_2 formation via the Criegee intermediate (CH_2OO) and cyclic compounds like dioxirane (CH_2OO^*) that is not generally considered in state-of-the-art empirical networks. In our view, domain knowledge and ML-based exploration should therefore be combined in practice.

Indeed, the generation of reference databases is also to an extent domain specific. The reaction networks considered herein are quite

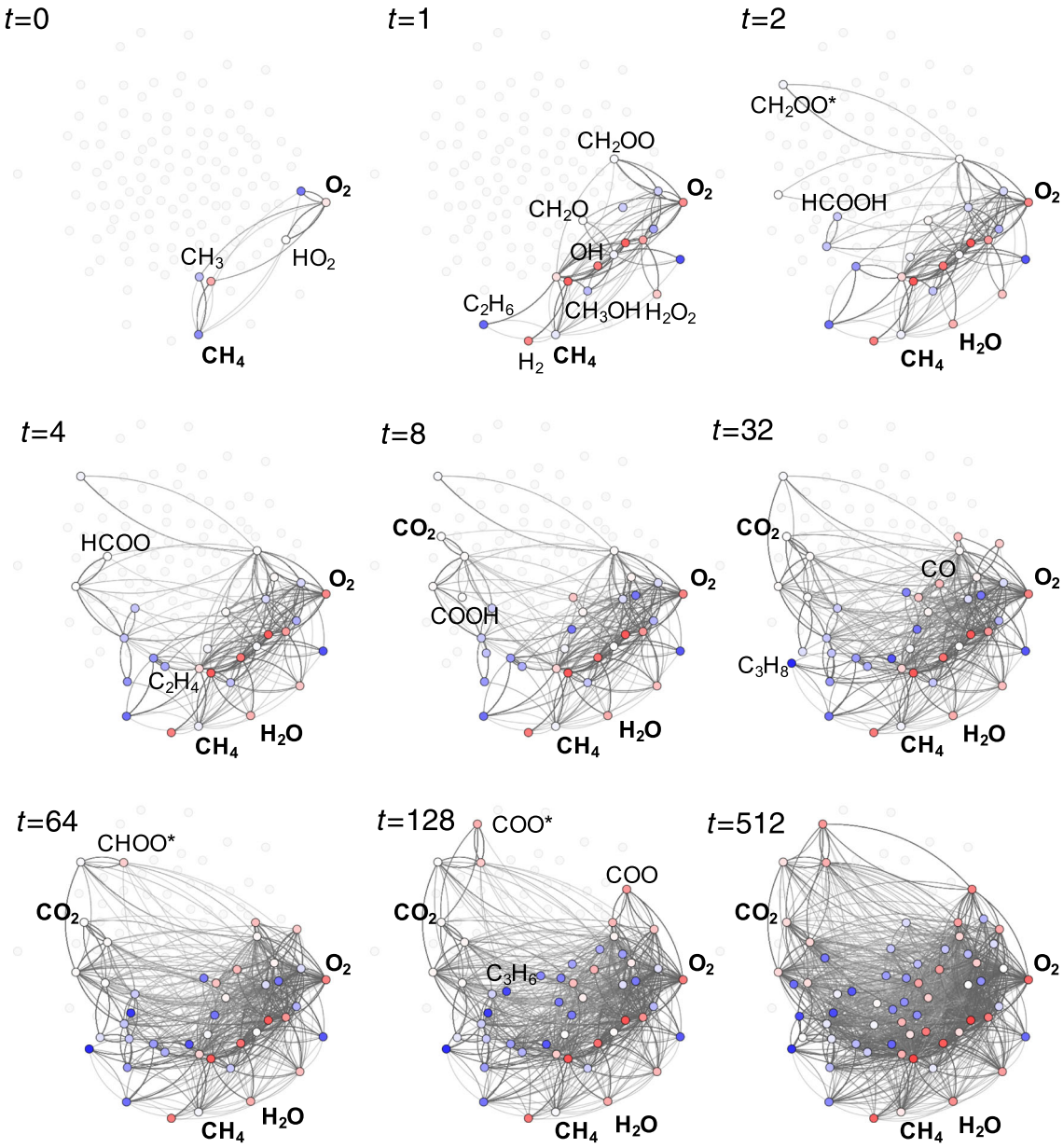


Fig. 7 ML-based exploration of a complex reaction network. Each frame shows the reduced reaction network extracted from a microkinetic simulation of methane combustion at different stages in simulation time. The abstract simulation time is shown for each frame in arbitrary units, see text. Educts and products (in bold), as well as important intermediates are highlighted. Nodes are colored according to their absolute atomization energies from low (red) to high (blue). Cyclic compounds are marked with an asterisk, to distinguish them from the corresponding linear compounds.

universal and could be applied to atmospheric chemistry, combustion or catalysis. However, these fields have distinct requirements with respect to the first-principles reference data. Clearly, catalysis can only be studied if the effect of the catalyst is accounted for. Meanwhile, thermal contributions to the free-energy will be large and important for a realistic description of combustion, and the role of different spin-states must be considered in both combustion and atmospheric chemistry. Nevertheless, the ML framework presented herein can easily be transferred to accommodate these situations.

To demonstrate this, a second set of energies for Rad-6 was computed using broken-symmetry (BS) DFT (see SI for details). In BS-DFT, the DFT energy is further minimized by exploiting the breaking of spatial and spin-symmetry in the Kohn-Sham determinant. The resulting determinants consequently do not correspond to a predefined multiplicity but represent the lowest energy solution irrespective of the spin state. Importantly, we find that ML models trained on this data have very similar predictive accuracy to the ones discussed so far (see Fig. 8). This shows that the Rad-6 database can serve as a benchmark for developing and improving ML models in reaction space, much like the popular QM9 set has done for chemical compound space.

Discussion

In this paper, we have explored the applicability of ML models to chemical reaction networks. In this context, we introduced the Rad-6 database of ca. 10,000 open and closed-shell molecules and an associated reaction network of ca. 30,000 reactions (Rad-6-RE). Established compound space KRR methods were shown to accurately predict atomization energies of the Rad-6 molecules. While the AE prediction accuracy was fairly similar for different choices in training set selection and kernel construction, these choices had a

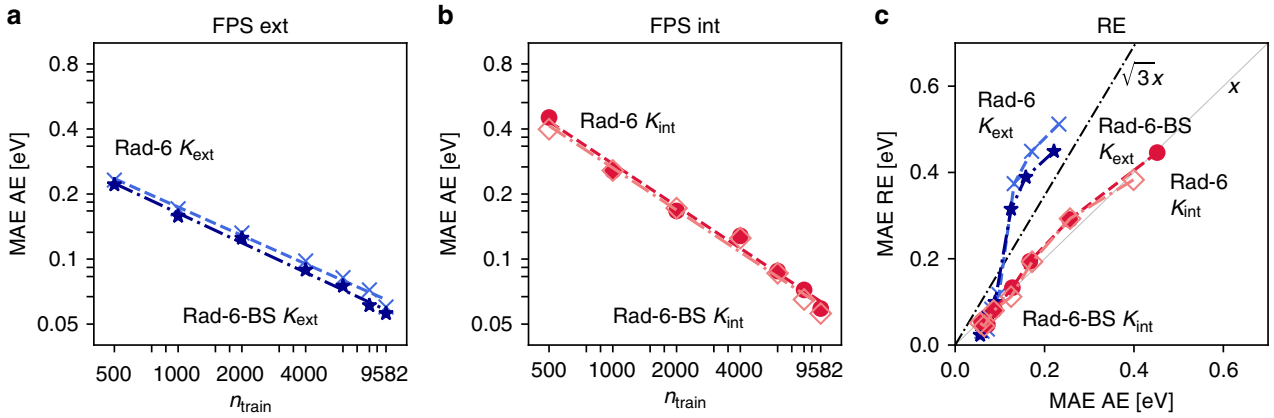


Fig. 8 Comparison ML models trained on the Rad-6 and the Rad-6-BS databases. a Learning curves for AE predictions of using the extensive kernel with an extensive FPS split and DFT geometries. **b** Same as (**a**) but for the intensive kernel with an intensive FPS split. **c** Correlation plot of MAE RE vs MAE AE for both Rad-6 and Rad-6-BS. Blue lines represent results obtained with the extensive kernel (crosses for Rad-6 and stars for Rad-6-BS) in (**a**) and (**c**). Red circles correspond to the intensive kernel with Rad-6 and orange diamonds to the intensive kernel with Rad-6-RE.

large effect on RE prediction accuracy. In particular, we found the use of an intensive kernel for both FPS-based training set selection and KRR learning to work very well for RE prediction, while models trained on extensive FPS sets displayed unexpectedly large RE errors. This can be rationalized by the special topology of reaction networks, in which certain small molecules constitute important hubs that should be included early on in the training sets.

We note that the extensive and intensive kernels used herein are merely interesting representatives of a wider range of possible models. Fundamentally, the observed differences in performance between the AE and RE prediction reflect that not all concepts established for the ML-based exploration of chemical compound space can be carried over to reaction space. Multiple methodological developments are required to establish reliable protocols, for example with respect to the weighting of molecules in the loss function of the ML model. If the topology of the reaction network of interest is known, these weights could for example be selected according to the connectivity of the molecule in the network (as shown in Fig. 1). Alternatively, weighting by size (or molecular weight) would likely be a useful heuristic to avoid the problems observed for the extensive kernel.

We also presented a proof-of-principle application of a reactive ML model to the exploration of the methane combustion reaction network. Here, a microkinetic simulation based on ML energetics was carried out, revealing relevant pathways and elementary steps in a large reaction network of 21,000 reactions. In our view, there are two ways to proceed from here. On one hand, the relevant subgraph thus extracted from of a much larger reaction network could be studied in depth with first-principles methods. On the other hand, we can envision an ML-driven computational reactor, where this is done in a more integrated fashion. Important steps (as identified by an ML-driven microkinetic simulation) could be studied with DFT and the results used to retrain the ML model. This would lead to an active-learning-type iterative procedure, where the predicted energetics of the reaction network are continuously improved in a targeted fashion, and no subgraph selection is necessary (within the computational constraints of the microkinetic simulation).

Methods

Computational details. Reference geometries and energies were obtained using DFT as implemented in `FHI-Aims`[46,71]. Specifically, the PBE0 functional[72] was used with tight integration settings and tier-2 numerical atomic orbital basis sets. Dispersion interactions were treated via the pair-wise Tkatchenko-Scheffler van-der-Waals correction[73]. Approximate geometries were obtained with the UFF forcefield[61].

Machine-learning models. All reported ML models are based on Kernel Ridge Regression and use the SOAP kernel[37,50]. SOAP representations were computed with the `quippy` code (https://github.com/libAtoms/QUIP). Kernel matrices and training/test splits were generated with the `mltools` package (https://github.com/simonwengert/mltools.git). The `atomic simulation environment` was used throughout to process molecular data[74].

Full methodological details are provided in the Supplementary information.

Data availability

All datasets used in this paper are available as Supplementary Data 1.

Code availability

The code used to fit the ML models is available at https://doi.org/10.5281/zenodo.4025972.

Received: 13 May 2020; Accepted: 1 October 2020;
Published online: 30 October 2020

References

1. Ulissi, Z. W., Medford, A. J., Bligaard, T. & Nørskov, J. K. To address surface reaction network complexity using scaling relations machine learning and DFT calculations. *Nat. Commun.* **8**, 14621 (2017).
2. Gossler, H., Maier, L., Angeli, S., Tischer, S. & Deutschmann, O. CaRMeN: an improved computer-aided method for developing catalytic reaction mechanisms. *Catalysts* **9**, 227 (2019).
3. Zhu, H., Kee, R. J., Janardhanan, V. M., Deutschmann, O. & Goodwin, D. G. Modeling elementary heterogeneous chemistry and electrochemistry in solid-oxide fuel cells. *J. Electrochem. Soc.* **152**, A2427 (2005).
4. Deutschmann, O. & Schmidt, L. D. Modeling the partial oxidation of methane in a short-contact-time reactor. *AIChE J.* **44**, 2465–2477 (1998).
5. Harper, M. R., Geem, K. M. V., Pyl, S. P., Marin, G. B. & Green, W. H. Comprehensive reaction mechanism for n-butanol pyrolysis and combustion. *Combust. Flame* **158**, 16–41 (2011).
6. Sankaran, R., Hawkes, E. R., Chen, J. H., Lu, T. & Law, C. K. Structure of a spatially developing turbulent lean methane-air bunsen flame. *Proc. Combust. Inst.* **31**, 1291–1298 (2007).
7. Smith, G. P. et al. Gri-mech 3.0. http://www.me.berkeley.edu/gri_mech/.
8. Vinu, R. & Broadbelt, L. J. Unraveling reaction pathways and specifying reaction kinetics for complex systems. *Annu. Rev. Chem. Biomol. Eng.* **3**, 29–54 (2012).
9. Vereecken, L., Glowacki, D. R. & Pilling, M. J. Theoretical chemical kinetics in tropospheric chemistry: methodologies and applications. *Chem. Rev.* **115**, 4063–4114 (2015).
10. Ashkenasy, G., Hermans, T. M., Otto, S. & Taylor, A. F. Systems chemistry. *Chem. Soc. Rev.* **46**, 2543–2554 (2017).
11. Grzybowski, B. A., Bishop, K. J. M., Kowalczyk, B. & Wilmer, C. E. The 'wired' universe of organic chemistry. *Nat. Chem.* **1**, 31–36 (2009).

12. Wächtershäuser, G. Evolution of the first metabolic cycles. *Proc. Natl Acad. Sci. USA* **87**, 200–204 (1990).
13. Simm, G. N. & Reiher, M. Systematic error estimation for chemical reaction energies. *J. Chem. Theory Comput.* **12**, 2762–2773 (2016).
14. Kowalik, M. et al. Parallel optimization of synthetic pathways within the network of organic chemistry. *Angew. Chem. Int. Ed.* **51**, 7928–7932 (2012).
15. Bajczyk, M. D., Dittwald, P., Wołos, A., Szymkuć, S. & Grzybowski, B. A. Discovery and enumeration of organic-chemical and biomimetic reaction cycles within the network of chemistry. *Angew. Chem. Int. Ed.* **57**, 2367–2371 (2018).
16. Bishop, K. J. M., Klajn, R. & Grzybowski, B. A. The core and most useful molecules in organic chemistry. *Angew. Chem. Int. Ed.* **45**, 5348–5354 (2006).
17. Fialkowski, M., Bishop, K. J. M., Chubukov, V. A., Campbell, C. J. & Grzybowski, B. A. Architecture and evolution of organic chemistry. *Angew. Chem. Int. Ed.* **44**, 7263–7269 (2005).
18. Simm, G. N. & Reiher, M. Context-driven exploration of complex chemical reaction networks. *J. Chem. Theory Comput.* **13**, 6108–6119 (2017).
19. Jacob, P.-M. & Lapkin, A. Statistics of the network of organic chemistry. *React. Chem. Eng.* **3**, 102–118 (2018).
20. Kim, Y., Kim, J. W., Kim, Z. & Kim, W. Y. Efficient prediction of reaction paths through molecular graph and reaction network analysis. *Chem. Sci.* **9**, 825–835 (2018).
21. Simm, G. N., Vaucher, A. C. & Reiher, M. Exploration of reaction pathways and chemical transformation networks. *J. Phys. Chem. A* **123**, 385–399 (2019).
22. Margraf, J. T. & Reuter, K. Systematic enumeration of elementary reaction steps in surface catalysis. *ACS Omega* **4**, 3370–3379 (2019).
23. Bruix, A., Margraf, J. T., Andersen, M. & Reuter, K. First-principles-based multiscale modelling of heterogeneous catalysis. *Nat. Catal.* **2**, 659–670 (2019).
24. Yang, N. et al. Intrinsic selectivity and structure sensitivity of rhodium catalysts for C_{2+} oxygenate production. *J. Am. Chem. Soc.* **138**, 3705–3714 (2016).
25. Medford, A. J. et al. Activity and selectivity trends in synthesis gas conversion to higher alcohols. *Top. Catal.* **57**, 135–142 (2014).
26. Yao, Z., Guo, C., Mao, Y. & Hu, P. Quantitative determination of C-C coupling mechanisms and detailed analyses on the activity and selectivity for Fischer-Tropsch synthesis on Co(0001): microkinetic modeling with coverage effects. *ACS Catal.* **9**, 5957–5973 (2019).
27. Rupp, M., Tkatchenko, A., Müller, K.-R. & von Lilienfeld, O. A. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.* **108**, 058301 (2012).
28. von Lilienfeld, O. A. Quantum machine learning in chemical compound space. *Angew. Chem. Int. Ed.* **57**, 4164–4169 (2018).
29. Bartók, A. P., Payne, M. C., Kondor, R. & Csányi, G. Gaussian approximation potentials: the accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.* **104**, 136403 (2010).
30. Behler, J. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *J. Chem. Phys.* **134**, 074106 (2011).
31. Hansen, K. et al. Assessment and Validation of Machine Learning Methods for Predicting Molecular Atomization Energies. *J. Chem. Theory Comput.* **9**, 3404–3419 (2013).
32. Stuke, A. et al. Chemical diversity in molecular orbital energy predictions with kernel ridge regression. *J. Chem. Phys.* **150**, 204121 (2019).
33. Häse, F., Valleau, S., Pyzer-Knapp, E. & Aspuru-Guzik, A. Machine learning exciton dynamics. *Chem. Sci.* **7**, 5139–5147 (2016).
34. Faber, F. A., Christensen, A. S., Huang, B. & von Lilienfeld, O. A. Alchemical and structural distribution based representation for universal quantum machine learning. *J. Chem. Phys.* **148**, 241717 (2018).
35. Huo, H. & Rupp, M. Unified representation of molecules and crystals for machine learning. https://arxiv.org/abs/2006.11223.
36. Bartók, A. P. et al. Machine learning unifies the modeling of materials and molecules. *Sci. Adv.* **3**, e1701816 (2017).
37. De, S., Bartók, A. P., Csányi, G. & Ceriotti, M. Comparing molecules and solids across structural and alchemical space. *Phys. Chem. Chem. Phys.* **18**, 13754–13769 (2016).
38. Stuke, A. et al. Atomic structures and orbital energies of 61,489 crystal-forming organic molecules. *Sci. Data* **7**, 58 (2020).
39. Unke, O. T. & Meuwly, M. PhysNet: a neural network for predicting energies, forces, dipole moments, and partial charges. *J. Chem. Theory Comput.* **15**, 3678–3693 (2019).
40. Schütt, K. T. et al. SchNetPack: a deep learning toolbox for atomistic systems. *J. Chem. Theory Comput.* **15**, 448–455 (2019).
41. Smith, J. S., Isayev, O. & Roitberg, A. E. ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chem. Sci.* **8**, 3192–3203 (2017).
42. Ruddigkeit, L., Van Deursen, R., Blum, L. C. & Reymond, J. L. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *J. Chem. Inf. Model.* **52**, 2864–2875 (2012).
43. Ramakrishnan, R. & von Lilienfeld, O. A. Many molecular properties from one kernel in chemical space. *Chim. Int. J. Chem.* **69**, 182–186 (2015).
44. Perdew, J. P., Ernzerhof, M. & Burke, K. Rationale for mixing exact exchange with density functional approximations. *J. Chem. Phys.* **105**, 9982–9985 (1996).
45. Tkatchenko, A. & Scheffler, M. Accurate Molecular Van Der Waals Interactions from Ground-State Electron Density and Free-Atom Reference Data. *Phys. Rev. Lett.* **102**, 073005 (2009).
46. Blum, V. et al. Ab initio molecular simulations with numeric atom-centered orbitals. *Comput. Phys. Commun.* **180**, 2175–2196 (2009).
47. Curtarolo, S. et al. The high-throughput highway to computational materials design. *Nat. Mater.* **12**, 191–201 (2013).
48. Andersen, M., Levchenko, S. V., Scheffler, M. & Reuter, K. Beyond scaling relations for the description of catalytic materials. *ACS Catal.* **9**, 2752–2759 (2019).
49. Jones, G., Bligaard, T., Abild-Pedersen, F. & Nørskov, J. K. Using scaling relations to understand trends in the catalytic activity of transition metals. *J. Phys. Condens. Matter* **20**, 064239 (2008).
50. Bartók, A. P., Kondor, R. & Csányi, G. On representing chemical environments. *Phys. Rev. B* **87**, 184115 (2013).
51. Willatt, M. J., Musil, F. & Ceriotti, M. Atom-density representations for machine learning. *J. Chem. Phys.* **150**, 154110 (2019).
52. Zuo, Y. et al. A performance and cost assessment of machine learning interatomic potentials. *J. Phys. Chem. A* **124**, 731–745 (2019).
53. Jung, H. et al. Size-extensive molecular machine learning with global representations. *ChemSystemsChem* **2**, e1900052 (2020).
54. Bishop, C. *Pattern Recognition and Machine Learning* (Springer, 2006).
55. Schölkopf, B., Smola, A. & Müller, K.-R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.* **10**, 1299–1319 (1998).
56. Cheng, B. et al. Mapping materials and molecules. *Acc. Chem. Res.* accepted (2020).
57. Goerigk, L. et al. A look at the density functional theory zoo with the advanced GMTKN55 database for general main group thermochemistry, kinetics and noncovalent interactions. *Phys. Chem. Chem. Phys.* **19**, 32184–32215 (2017).
58. Margraf, J. T., Ranasinghe, D. S. & Bartlett, R. J. Automatic generation of reaction energy databases from highly accurate atomization energy benchmark sets. *Phys. Chem. Chem. Phys.* **19**, 9798–9805 (2017).
59. Perdew, J. P., Sun, J., Garza, A. J. & Scuseria, G. E. Intensive atomization energy: re-thinking a metric for electronic structure theory methods. *Z. Phys. Chem.* **230**, 737–742 (2016).
60. Rupp, M. Machine learning for quantum mechanics in a nutshell. *Int. J. Quantum Chem.* **115**, 1058–1073 (2015).
61. Rappe, A. K., Casewit, C. J., Colwell, K. S., Goddard, W. A. & Skiff, W. M. UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations. *J. Am. Chem. Soc.* **114**, 10024–10035 (1992).
62. Grimme, S., Bannwarth, C. & Shushkov, P. A robust and accurate tight-binding quantum chemical method for structures, vibrational frequencies, and noncovalent interactions of large molecular systems parametrized for all spd-block elements (Z = 1-86). *J. Chem. Theory Comput.* **13**, 1989–2009 (2017).
63. Gaus, M., Goez, A. & Elstner, M. Parametrization and benchmark of DFTB3 for organic molecules. *J. Chem. Theory Comput.* **9**, 338–354 (2013).
64. Mezei, P. D. & von Lilienfeld, O. A. Noncovalent quantum machine learning corrections to density functionals. *J. Chem. Theory Comput.* **16**, 2647–2653 (2020).
65. Bagheri, G. et al. Comprehensive kinetic study of combustion technologies for low environmental impact: MILD and OXY-fuel combustion of methane. *Combust. Flame* **212**, 142–155 (2020).
66. Onda, T., Nakamura, H., Tezuka, T., Hasegawa, S. & Maruta, K. Initial-stage reaction of methane examined by optical measurements of weak flames in a micro flow reactor with a controlled temperature profile. *Combust. Flame* **206**, 292–307 (2019).
67. Hu, F. et al. Global reaction mechanisms for MILD oxy-combustion of methane. *Energy* **147**, 839–857 (2018).
68. Chu, T.-C. et al. Modeling of aromatics formation in fuel-rich methane oxy-combustion with an automatically generated pressure-dependent mechanism. *Phys. Chem. Chem. Phys.* **21**, 813–832 (2019).
69. Si, J., Wang, G., Li, P. & Mi, J. Optimization of the global reaction mechanism for MILD combustion of methane using artificial neural network. *Energy Fuels* **34**, 3805–3815 (2020).
70. Laguillo, S., Ochoa, J. S. & Ortiz, A. Chemical reaction mechanisms assessment for simulation of methane combustion in domestic gas cooking burners. *Energy Fuels* **33**, 9171–9183 (2019).

71. Zhang, I. Y., Ren, X., Rinke, P., Blum, V. & Scheffler, M. Numeric atom-centered-orbital basis sets with valence-correlation consistency from H to Ar. *N. J. Phys.* **15**, 123033 (2013).
72. Adamo, C. & Barone, V. Towards reliable density functional methods without adjustable parameters: the PBE0 model. *J. Chem. Phys.* **110**, 6158–6170 (1999).
73. Tkatchenko, A., DiStasio, R. A., Car, R. & Scheffler, M. Accurate and efficient method for many-body van der waals interactions. *Phys. Rev. Lett.* **108**, 236402 (2012).
74. Hjorth Larsen, A. et al. The atomic simulation environment-a Python library for working with atoms. *J. Phys. Condens. Matter* **29**, 273002 (2017).

Acknowledgements

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) through TUM International Graduate School of Science and Engineering (IGSSE) (GSC 81) and by the TUM Institute for Advanced Study, which awarded a August-Wilhelm-Scheer visting professorship to G.C. We gratefully acknowledge S. Wengert for the technical support via mltools and for fruitful discussions.

Author contributions

S.S. performed the DFT calculations on the Rad-6 database and fitted the ML models. J.T.M. performed the reaction network analysis and the BS-DFT calculations. G.C., K.R., and J.T.M. devised the project. All authors contributed to writing the manuscript.

Funding

Open Access funding enabled and organized by Projekt DEAL.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41467-020-19267-x.

Correspondence and requests for materials should be addressed to J.T.M.

Peer review information *Nature Communications* thanks Carl Simon Adorf, Jan Halborg Jensen and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Supporting Information - Machine Learning in Chemical Reaction Space

Sina Stocker,[1] Gábor Csányi,[2] Karsten Reuter,[1,3] and Johannes T. Margraf[1]

[1)] *Chair of Theoretical Chemistry and Catalysis Research Center, Technische Universität München, Garching, Germany*

[2)] *Engineering Laboratory, University of Cambridge, Cambridge CB2 1PZ, United Kingdom*

[3)] *Fritz-Haber-Institut der Max-Planck-Gesellschaft, Berlin, Germany*

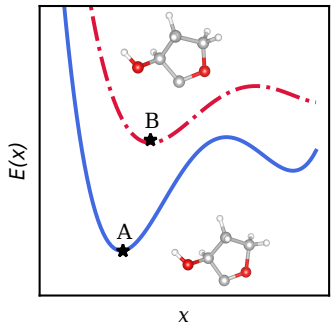
(Dated: 25 September 2020)

Supplementary Note 1: Rad-6 Database

The Rad-6 reference database comprises both closed-shell molecules and (poly-)radical fragments containing carbon, oxygen and hydrogen (see below for a detailed description). SMILES strings[1] for structures containing up to 6 non-hydrogen atoms were created using the graph-based approach of Margraf and Reuter[2] and subsequently converted to 3D structures using the RDKit package.[3] Geometries were initially relaxed with the universal forcefield (UFF).[4] Final geometries and energies were obtained using DFT as implemented in FHI-Aims[5,6]. Specifically, the PBE0 functional[7] was used with tight integration settings and tier-2 numerical atomic orbital basis sets. Dispersion interactions were treated via the pair-wise Tkatchenko-Scheffler van-der-Waals correction.[8] The final reported geometries are converged to a maximum residual force component of $10\,\mathrm{meV\,Å^{-1}}$ per atom.

As ML models do not explicitly consider electronic structure, special considerations with respect to spin states are required. In Rad-6, all DFT calculations were initialized with low-spin densities (singlet multiplicity for even number of electrons, doublet for odd number of electrons), constructed according to the location of radical electrons in the SMILES string. Exceptions were made for the carbon and oxygen atoms as well as for the oxygen molecule, which were treated as triplets. This was to ensure correct atomization energies and reasonable energetics for oxidation reactions with O_2. All open-shell systems were treated with collinear spin-polarization. These choices are arbitrary, but inconsequential to the conclusions of this study. A rigorous treatment of spin in a ML context is in principle possible, but this would require fitting a separate model for each spin-state.

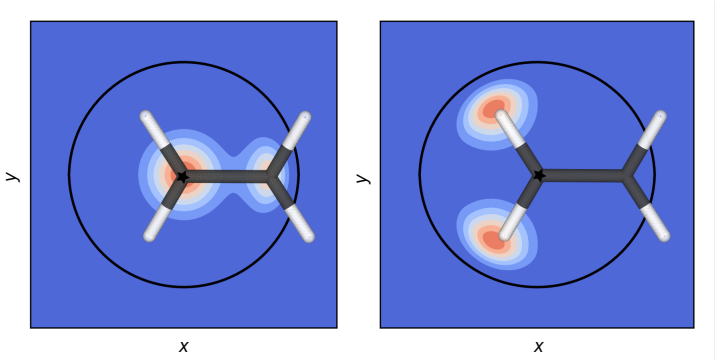
A second important issue relates to the geometries used for ML. Many of the initially constructed poly-radicals decompose during the DFT relaxation or simply do not converge. This is problematic for two reasons: Firstly, the definition of chemical reactions presupposes a certain molecular topology (i.e. how atoms are connected). Secondly, in this case the UFF geometry (with fixed topology) describes a different molecule than the DFT one. In a realistic setting, the DFT geometries will not be available for ML predictions. If they were, the DFT energy would also be known and the ML prediction would be redundant.[9,10] In the main manuscript, both UFF and DFT geometries are used for training and prediction, but the energies of relaxed DFT geometries are always the target property (see Supplementary Figure 1).



Supplementary Figure 1. Schematic representation of a DFT (blue, solid line) and forcefield (red, dashed line) potential energy surface. As it is mentioned in the text, two different types of ML models are used in this work. (1) The ML models predict the energies of relaxed DFT geometries (point A) and the corresponding DFT geometries (point A). (2) The ML models predict the energies of relaxed DFT geometries (point A) based on structures relaxed with a forcefield (point B).

In order to allow for a clear definition of reactions in terms of bond-breaking and formation, only those systems where UFF and DFT geometries describe the same molecular topology are included in the database. This leads to a drastic reduction from the initial set of over 27,000 systems to 10,712 structures in the final Rad-6 database. A positive side effect of this is that the structures in Rad-6 can be expected to be reasonably stable, since they represent local minima on the DFT calculated potential energy surface.

Rad-6-BS database: To investigate the stability of the proposed ML approach with respect to changes in the data (in particular regarding the spin-state), a second set of single-point energies was calculated for the Rad-6 database, using broken-symmetry DFT. Here, calculations were performed with the revPBE functional and def2-TZVP basis-set using Orca.[11,12] To enable symmetry breaking even for nominally closed-shell systems, the beta-spin orbitals in the initial guesses were perturbed by randomly mixing an occupied and unoccupied orbital. After convergence, four additional calculations were performed for each system, reusing the converged wavefunctions from previous runs and further perturbing the orbitals. This procedure was used to avoid SCF convergence into local minima or saddle points.[13]



Supplementary Figure 2. The smooth overlap of atomic positions (SOAP) kernel uses a three-dimensional neighborhood density function $\rho_a^Z(\mathbf{r})$ of broadened atomic positions within a cutoff. As mentioned in the text, for every species a separate density is constructed. Planar cuts through $\rho_a^{\mathrm{C}}(\mathbf{r})$ (left) and $\rho_a^{\mathrm{H}}(\mathbf{r})$ (right) around a cutoff centered carbon atom (star) in ethylene are shown. The black circle represents the radial cutoff distance.

Supplementary Note 2: Theory and Computational Methods

Smooth overlap of atomic positions (SOAP): SOAP is a local kernel that measures the similarity of atomic environments.[10] It was found to be highly successful in molecular and solid-state applications.[10,14–16] Below, a brief overview of the concept is given, more details can be found in the literature.[17,18]

SOAP is based on the neighborhood density function $\rho_a(\mathbf{r})$ around a reference atom a:

$$\rho_a(\mathbf{r}) = \sum_{i \in \chi_a} \exp\left(-\frac{(\mathbf{r} - r_{ai})^2}{2\sigma_{\mathrm{at}}^2}\right) \times f_{cut}(\mathbf{r}) \tag{1}$$

where the sum runs over all neighboring atoms i (within a cutoff radius, the atomic environment χ) and $f_{cut}(\mathbf{r})$ is a damping function ensuring that the density smoothly approaches zero at the cutoff. Each atom (including the reference atom) within the cutoff is broadened with a Gaussian of width σ_{at}, leading to a smooth, local representation of the atomic environment.

The atom centred neighborhood density in Supplementary Eq. 1 complies with a system containing only one type of atomic species. For systems with different types of elements, like molecules, the density is individually constructed for every atomic species (Z) within

the atomic environment χ of atom a (see Supplementary Figure 2):

$$\rho_a^Z(\mathbf{r}) = \sum_{i \in \chi_a^Z} \exp\left(-\frac{(\mathbf{r} - r_{ai})^2}{2\sigma_{\text{at}}^2}\right) \times f_{cut}(\mathbf{r}). \tag{2}$$

The similarity between two such environments can be measured via a rotationally averaged overlap integral:

$$\tilde{k}(\chi_a, \chi_b) = \int d\hat{R} \left| \int \sum_Z \rho_a^Z(\mathbf{r}) \rho_b^Z(\hat{R}\mathbf{r}) d\mathbf{r} \right|^2 \tag{3}$$

where the outer integral is over all rotations $\hat{R}$, so that $\tilde{k}(\chi_a, \chi_b)$ is invariant to rotations or permutations of atoms. The power of two in the inner integral ensures that the kernel retains angular information about the neighborhood density.

Importantly, this integral can be solved analytically, if the neighborhood density is expanded in an atom-centered basis of orthogonal radial basis functions $g_n(|\mathbf{r}|)$ and spherical harmonics Y_{lm}:

$$\rho_{\chi_a}^Z(\mathbf{r}) = \sum_{nlm} c_{nlm}^Z g_n(|\mathbf{r}|) Y_{lm}(\mathbf{r}). \tag{4}$$

The coefficients c_{nlm}^Z are then transformed into the so-called power spectrum for individual species:

$$\mathbf{p}_{nn'l}^{Z_1 Z_2} = \pi \sqrt{\frac{8}{2l+1}} \sum_m (c_{nlm}^{Z_1})^\dagger c_{n'lm}^{Z_2} \tag{5}$$

which we truncate at n $\leq$ 8 and l $\leq$ 8.

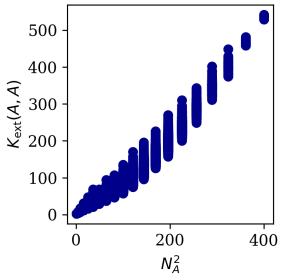
The kernel from Supplementary Eq. 3 can now be computed as a simple dot product of the 'partial' power spectra:

$$\tilde{k}(\chi_a, \chi_b) = \sum_{Z_1 Z_2} \mathbf{p}_{Z_1 Z_2}(\chi_a) \mathbf{p}_{Z_1 Z_2}(\chi_b) \tag{6}$$

To obtain the final SOAP kernel, this function is normalized and squared so that:

$$k(\chi_a, \chi_b) = \left(\frac{\tilde{k}(\chi_a, \chi_b)}{\sqrt{\tilde{k}(\chi_a, \chi_a)\tilde{k}(\chi_b, \chi_b)}} \right)^2. \tag{7}$$

The atomic kernels are the basis to build global kernels for structure matching (e.g. molecules) instead of local environments. Here, we use the average (Eq. 4) and sum kernel



Supplementary Figure 3. Kernel diagonal elements against the number of atoms in a molecule squared. The plot shows that $K_{\text{ext}}(A, A) \sim N_A^2$.

(Eq. 5) described in the main text. It is worth stressing that the average kernel, an intensive kernel has to be normalized:

$$K(A, B) = \frac{\bar{K}(A, B)}{\sqrt{\bar{K}(A, A)\bar{K}(B, B)}}, \tag{8}$$

while the sum kernel should not, i.e. $K(A, B) = K^{\Sigma}(A, B)$. Consequently, the magnitude of the diagonal elements of the sum kernel matrix scales with the square of the number of atoms in the molecule (see Supplementary Figure 3)

In this work, we use the `quippy` code[19] and the `mltools` package[20] to compute SOAP kernels. In order to have flexibility in the description of short and mid-range contributions, a kind of 'multiscale' (ms) SOAP is used. Specifically, two global SOAP kernels with cutoff values of 2 Å (K_2) and 4 Å (K_4) are applied simultaneously. We use σ_{at} of 0.3 Å for K_2 and σ_{at} of 0.6 Å for K_4. We combine short and mid-range contributions for the average kernel as the average of the normalized kernels K_2 and K_4, i.e. $K^{\text{ms}}_{\text{int}} = \frac{K_{2,\text{int}}+K_{4,\text{int}}}{2}$. For the sum kernel we simply sum up the individual sum kernels $K^{\text{ms}}_{\text{ext}} = K_{2,\text{ext}} + K_{4,\text{ext}}$.[21]

Kernel ridge regression: Kernel ridge regression (KRR) is a supervised machine learning technique to obtain function values for given input configurations x_i. In this section we give a short overview about this technique, however for a detailed description and mathematical derivations the reader is referred to literature.[22]

The function can be expressed as linear combinations of kernel functions ($K(x_i, x)$):

$$f(x) = \sum_i^N \alpha_i K(x_i, x), \tag{9}$$

while the kernel functions act as similarity measures between different input configurations x and x_i with target properties y and y_i. The x_i are feature vectors of training data

points and α_i are regression weights.

KRR provides a closed-form solution for the optimal set of weights α. This can be obtained by minimizing the loss-function l (of a regularized least-squares problem):

$$l = \sum_{j}^{N} \left(\sum_{i}^{N} \alpha_i K(x_i, x_j) - y_j \right)^2 + \sigma \alpha^T \mathbf{K} \alpha \tag{10}$$

The solution of this problem is than given in matrix vector notation:

$$\alpha = (\mathbf{K} + \sigma \mathbf{I})^{-1} \mathbf{y}, \tag{11}$$

where $\mathbf{K}$ is the kernel matrix of the training set (with $K_{ij} = K(x_i, x_j)$), σ is the regularization parameter and $\mathbf{I}$ is the identity matrix. σ is a hyperparameter that has to be determined empirically (see Supplementary Note 3). It represents the noise level of the reference data and is used to control over- and underfitting.

In our work we applied mean-correction to the observables in the fit with the intensive kernel while we did not for the extensive kernel.

Kernel principal component analysis (kPCA): Principal component analysis is a tool for projecting high dimensional data into a lower dimensional space and therefore enables the visualization of that specific data. In PCA, data is transformed into a new coordinate system such that the new coordinate axes point into the direction of largest variance (first coordinate into the direction of largest variance, so-called PC 1, second coordinate into the direction of second largest variance and orthogonal to PC 1, so-called PC 2, ...). Kernel PCA is an extension to PCA and makes the dimensionality reduction of non-linear data possible.[23]

To this end, the kernel matrix is constructed analogous to KRR and then 'centralized':

$$\hat{\mathbf{K}} = \mathbf{K} - \mathbf{1}_\mathrm{N}\mathbf{K} - \mathbf{K}\mathbf{1}_\mathrm{N} + \mathbf{1}_\mathrm{N}\mathbf{K}\mathbf{1}_\mathrm{N}, \tag{12}$$

where $\mathbf{1}_\mathrm{N}$ is a matrix with the same dimensions as the kernel matrix, in which every element is identically $1/N$ (with the number of data points N). For $\hat{\mathbf{K}}$, the eigenvalue problem has to be solved,

$$\hat{\mathbf{K}}\mathbf{v}_i = \lambda_i \mathbf{v}_i, \tag{13}$$

where $\mathbf{v}_i$ is the i^{th} eigenvector and λ_i the respective eigenvalue. The data can be projected into the new space via:

$$\mathbf{PC}_i = \mathbf{K}\mathbf{v}_i. \tag{14}$$

Supplementary Note 3: Training Set Selection, Hyperparameter Search and Learning Curves

Training set selection: We divide the Rad-6 database into a training, validation (100 structures) and test set (1030 structures). The farthest point sampling (FPS) technique is used to select representative and diverse training configurations. The FPS algorithm starts with an arbitrary data point and sequentially adds new structures so that the distance between the newest structure and all previously selected ones is maximized.[10,18,21] This requires a distance matrix that is constructed using the kernel, according to:

$$D(A,B) = \sqrt{(K(A,A) + K(B,B) - 2K(A,B))} \tag{15}$$

A sequence is generated for the complete Rad-6 database. The last 1030 structures went into test set and 100 structures before the last 1030 into the validation set. Since the distance matrix is a function of the kernel, we obtain different training, validation and test sets for the average and sum kernel. In this work the FPS is done with K_{int} and K_{ext} using UFF geometries and started with the H-atom, respectively.

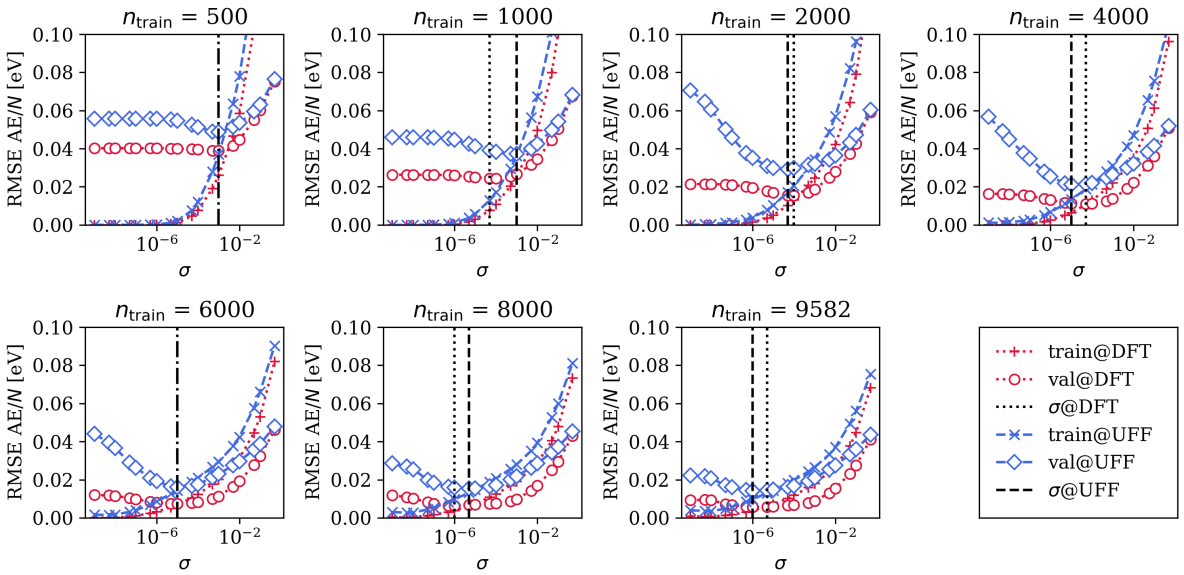
Hyperparameter search: Our ML models contain several hyperparameter in the SOAP kernel and one hyperparameter in kernel ridge regression. In this work we do not focus on the optimization of the hyperparameter in the SOAP kernel, but optimize the σ hyperparameter in KRR. This is done by evaluating the RMSE of the validation set in a grid search.[9] The results for all kernels and FPS splits are shown in Supplementary Figures 4-7.

Including also the hyperparameter optimization for the SOAP kernels could lead to even smaller errors on the predictions.

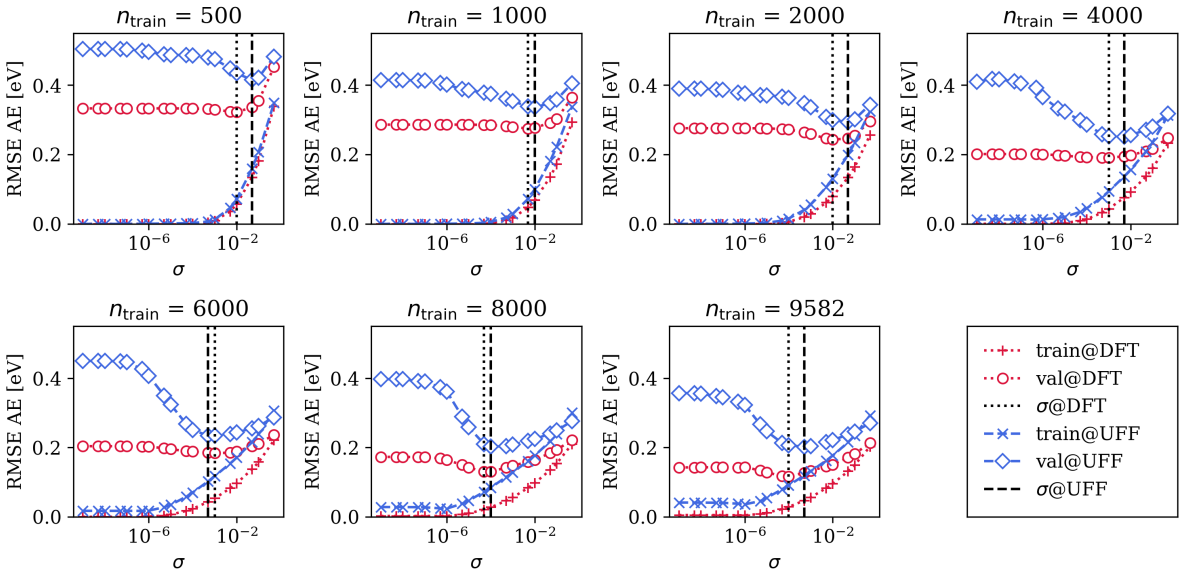
Learning curves: Learning curves of AE and RE for the respective kernels and FPS splits are shown in Supplementary Figures 8-11. These plots show the MAE and RMSE for training, validation and test set (two left subplots) as well as for the reaction network Rad-6-RE (two right subplots).

Supplementary Note 4: Learning AE with UFF Geometries

Supplementary Figure 12 displays the results for the predictions of atomization energies using DFT geometries (Fig. 4 main text) as well as the MAEs for AE using UFF geometries. As mentioned in the main text, using UFF instead of DFT geometries leads to the same

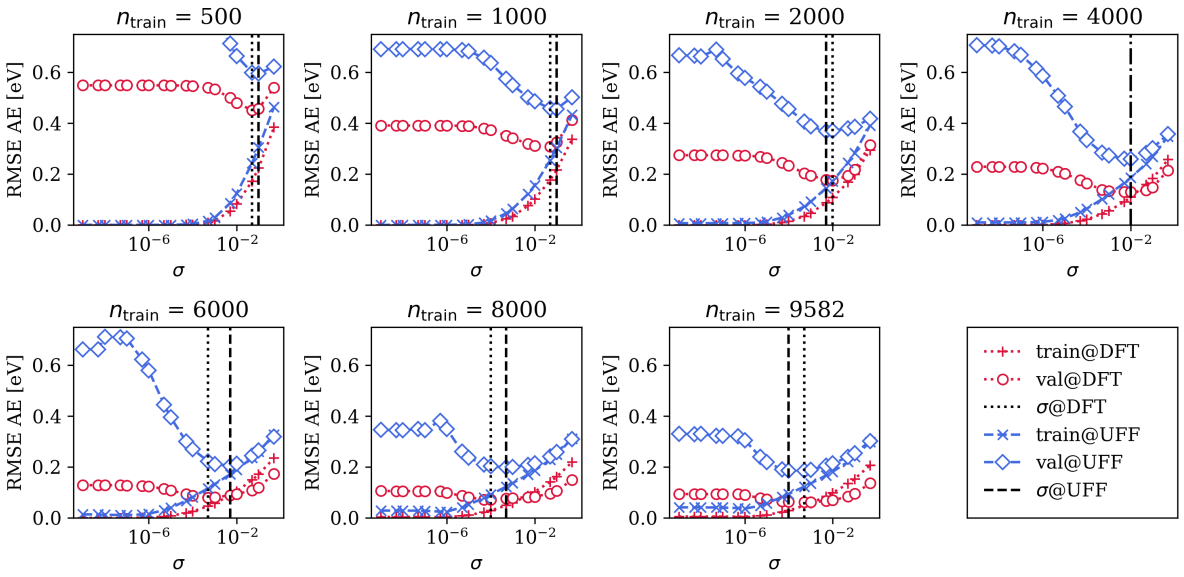


Supplementary Figure 4. Hyperparameter search for K_{int} FPS int.

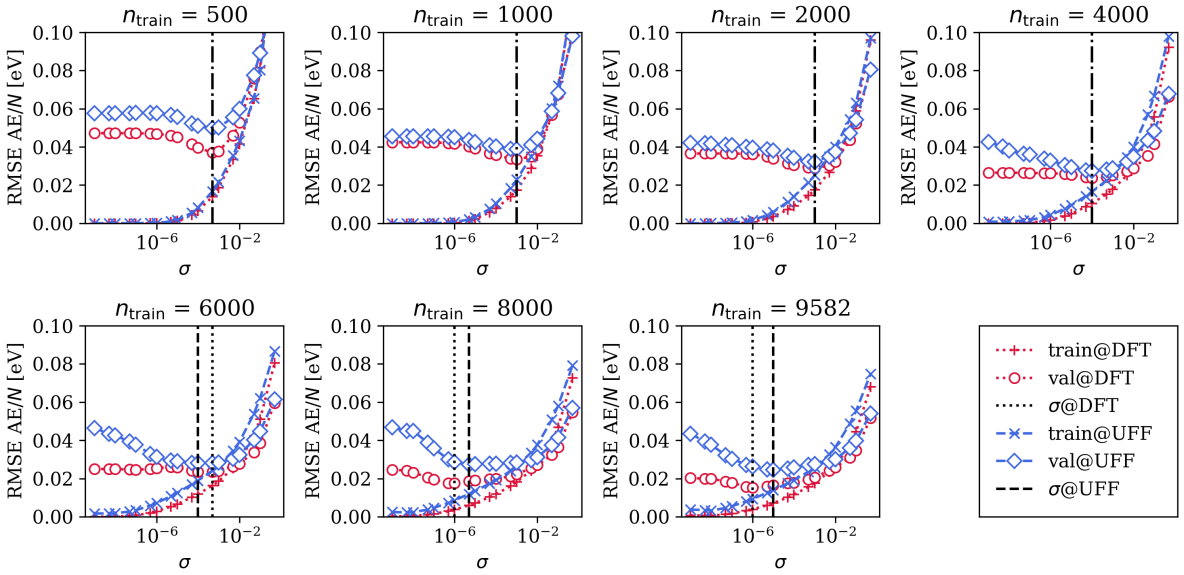


Supplementary Figure 5. Hyperparameter search for K_{ext} FPS ext.

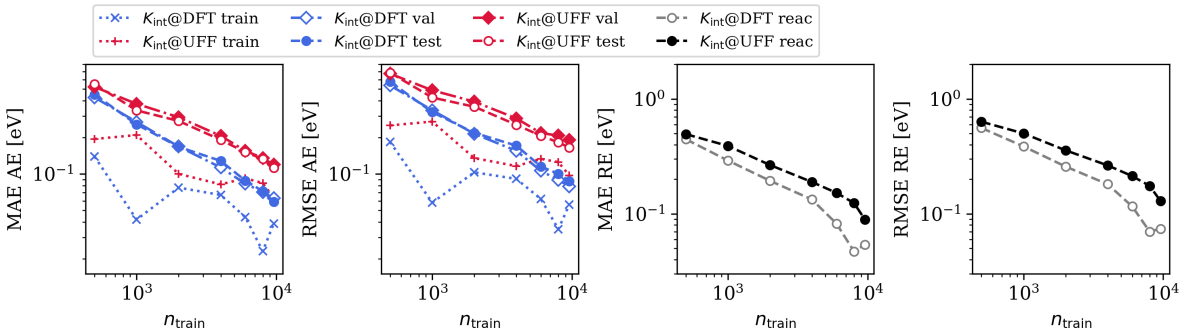
trends in learning curves for the different kernels and FPS splits. However, it results in higher errors on the predictions since the ML model has to additionally learn the differences between the geometries for the different levels of theory (see. Supplementary Figure 1).



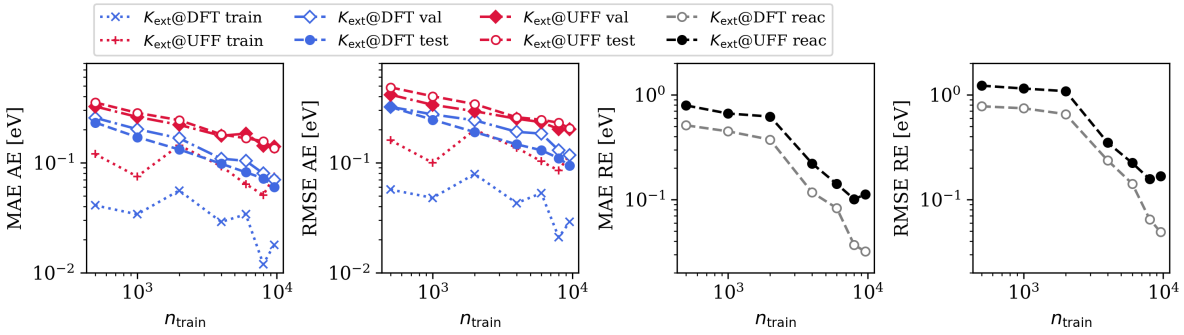
Supplementary Figure 6. Hyperparameter search for K_{ext} FPS int.



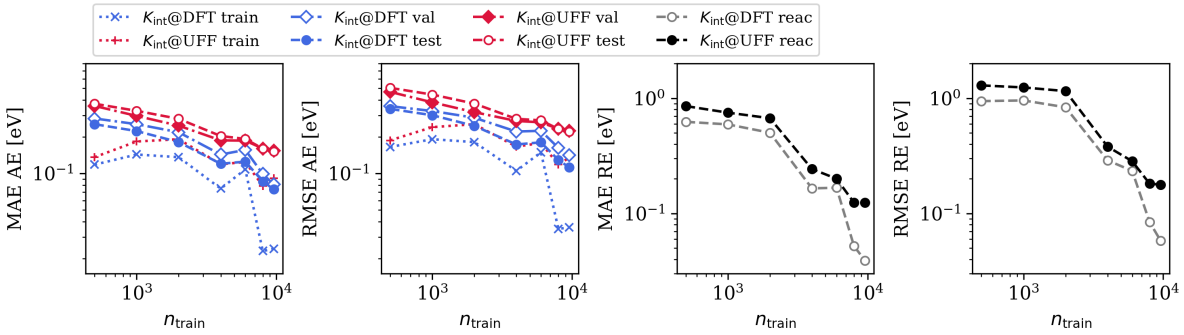
Supplementary Figure 7. Hyperparameter search for K_{int} FPS ext.



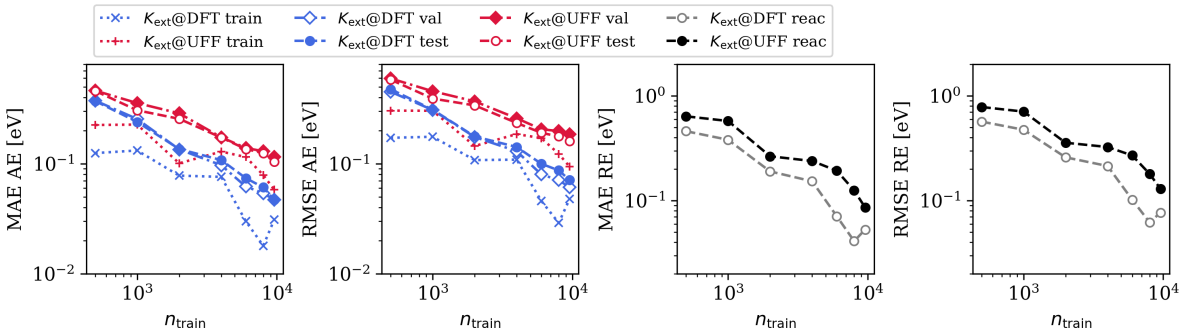
Supplementary Figure 8. Learning curves K_{int} FPS int.



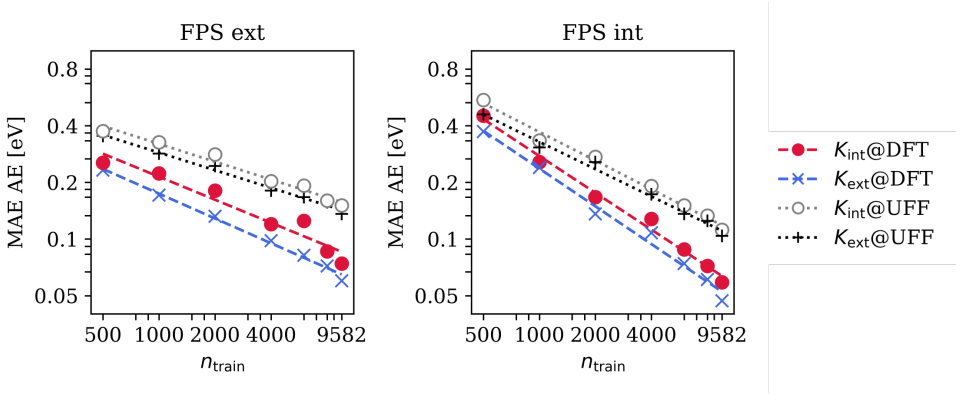
Supplementary Figure 9. Learning curves K_{ext} FPS ext.



Supplementary Figure 10. Learning curves K_{int} FPS ext.



Supplementary Figure 11. Learning curves K_{ext} FPS int.



Supplementary Figure 12. Learning curves for atomization energy (AE) predictions (on the test set) using extensive and intensive kernels and DFT and UFF geometries. The two subplots show the results for both FPS splits.

Supplementary Note 5: Timings of ML Model vs. DFT Calculations

A fundamental advantage of using ML is that the predictions for new data points can be made in much less time than the original calculations. To illustrate this we provide the timings for 100 predicitons on random molecules from the Rad-6 database for (1) DFT calculations using computational settings listed in Supplementary Note 1 and (2) a KRR ML model trained on 9582 configurations. More precisely the recorded time for the ML model refers to the calculation of the multisoap average kernel, i.e. the generation of two 9582×100 matrices (K_2 and K_4, see Supplementary Note 2) and the prediction of the 100 molecules using the previously obtained model coefficients α. Unsurprisingly, the KRR model is more than two order of magnitudes faster than a geometry optimization at DFT (PBE0) level.

Supplementary Table 1. Comparison of timings for AE prediction of 100 molecules with the ML model and via full DFT geometry relaxation (at the PBE0+TS level). More details are given in the text.

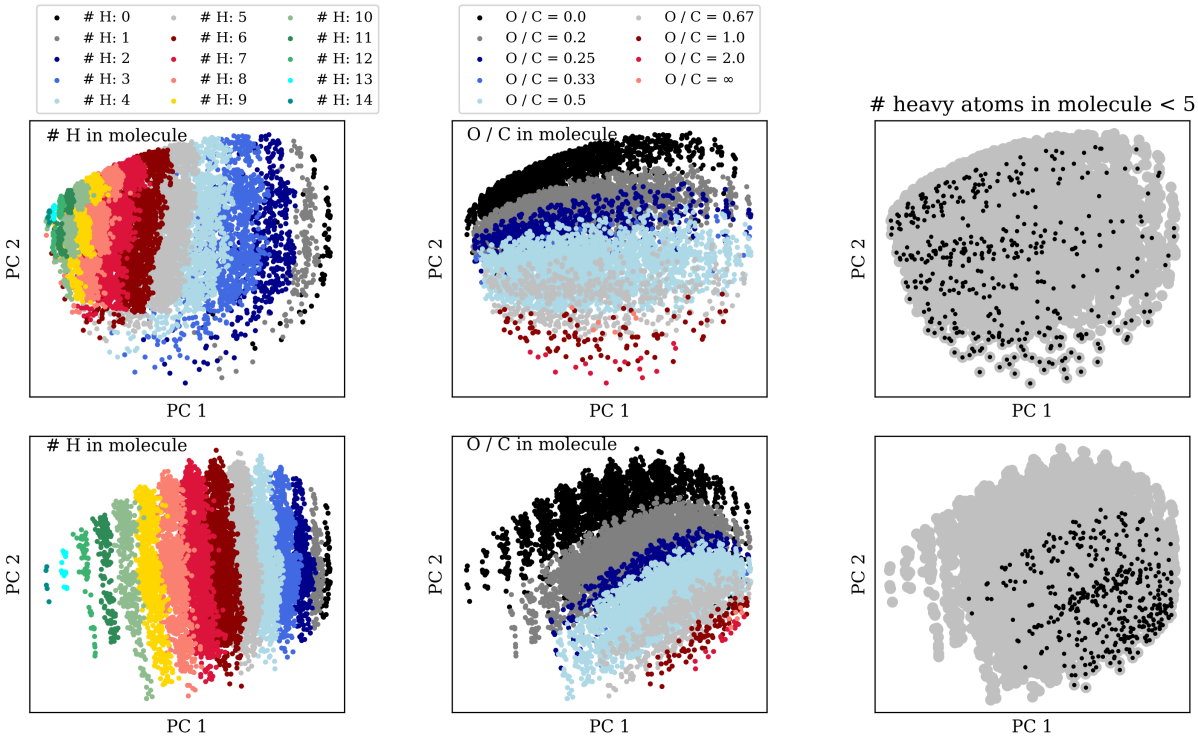
	DFT	KRR
total time used	949.03 h	1468.19 s
time per molecule	9.49 h	14.68 s

Supplementary Note 6: kPCA

kPCA is a data visualization tool in which huge data sets are intuitively presented and insights into the database are provided. Herein, kPCA is used to have a closer look into the Rad-6 database and visualize similarities and differences between the intensive and extensive kernel (see Supplementary Figures 13, 14).

The location of molecules in the PCA plot is determined by their structural topology. Specifically, PC 1 separates saturated molecules (like hexane) on the left in the PCA plot from very unsaturated ones (like fumaryl) on the right. Simply put, the separation of molecules among PC 1 results in counting hydrogen atoms in the molecules. This is slightly more pronounced in the extensive kernel visible in the colored stripes in Supplementary Figure 13. Furthermore, PC 2 displays the ratio of O / C atoms in the molecule.

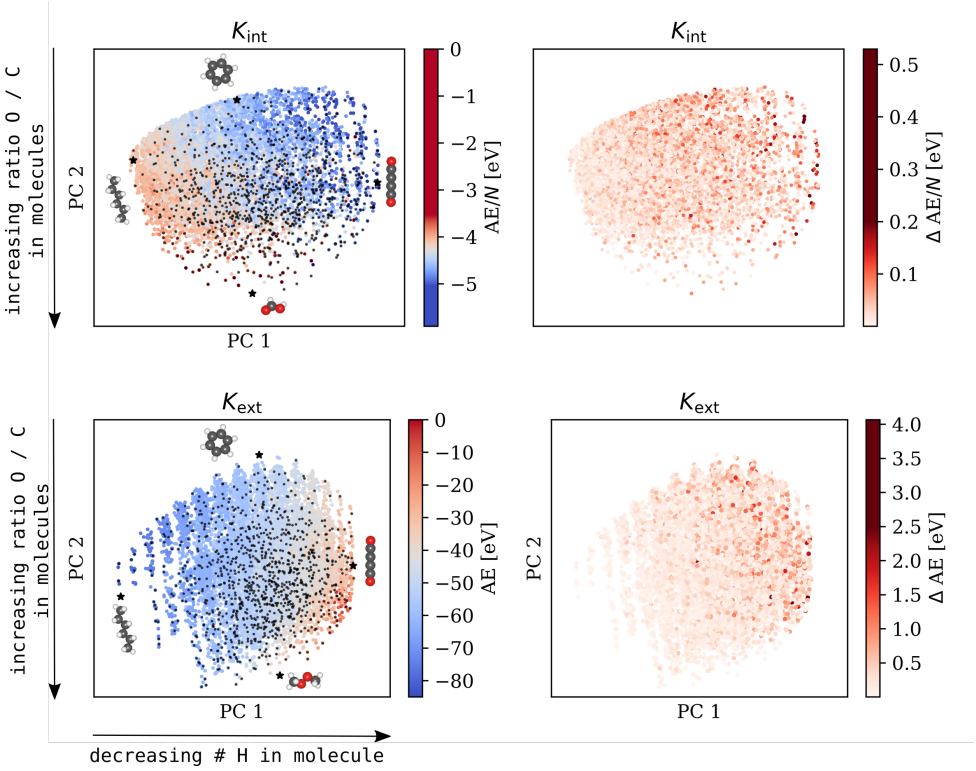
The upper and lower right sub-panels in Supplementary Figure 13 show the distribution



Supplementary Figure 13. kPCA plots of molecules based on the intensive (top) and extensive (bottom) kernel using UFF geometries. Left column: Separation of molecules through PC 1. Colors represent the number of H atoms in a molecule. Middle column: Separation of molecules through PC 2. Colors represent the O / C ratio in a molecule. Right column: Distribution of small molecules with maximum 4 heavy atoms in a molecule.

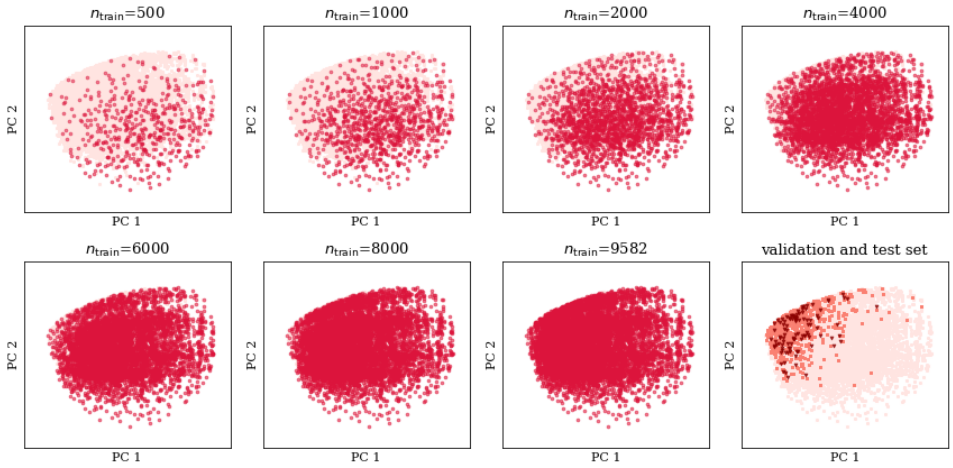
of small molecules located in the database. We denote molecules with a maximum number of 4 heavy atoms (i.e. non H-atoms) as small molecules. These account for around 4 % of the database. While the small molecules are distributed over the whole space in the plot of K_{int}, for K_{ext} they are bounded on the bottom right. This picture illustrates why small molecules are selected relatively late in FPS with the extensive kernel, because the distances among them are relatively close.

Supplementary Figure 14 is an extension of Fig. 3 in the main text. The plot shows the kPCA for the extensive (bottom) and intensive (top) kernels colored by the predicted atomization energies and atomization energies per atom for ML models with 1000 training configurations, respectively. Additionally, the differences between the ML models and the reference DFT calculated energies are displayed on the right. K_{ext} shows higher errors on

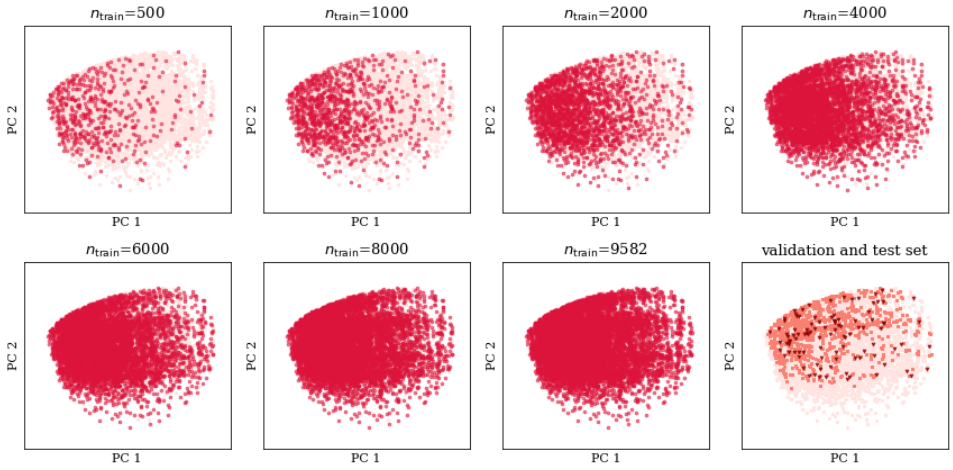


Supplementary Figure 14. kPCA plot of molecules based on the intensive (top) and extensive (bottom) kernel using UFF geometries: Points are colored according to the predicted atomization energy per atom (top left) and the predicted total atomization energy (bottom left) using ML models with 1000 training points. The absolute differences between the ML models and DFT reference values are shown on the top right picture for K_{int} and on the bottom right for K_{ext}. Small black dots indicated training structures. The arrows provide a qualitative interpretation of the principal component axes.

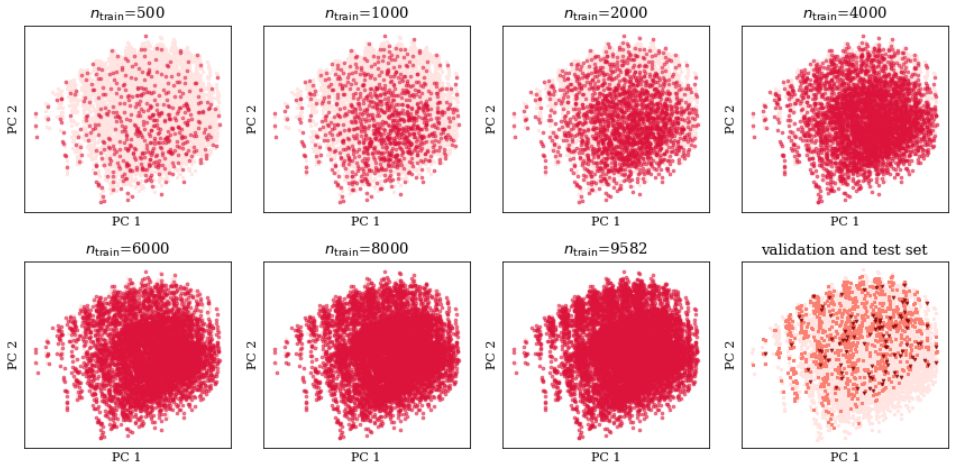
the predictions on the right half of the plot where especially small molecules are located. This illustrates again the poor performance of the extensive kernel in predicting reaction energies. A precise description of small molecules is crucial for calculating reaction energies, since they represent important hubs in the reaction network.



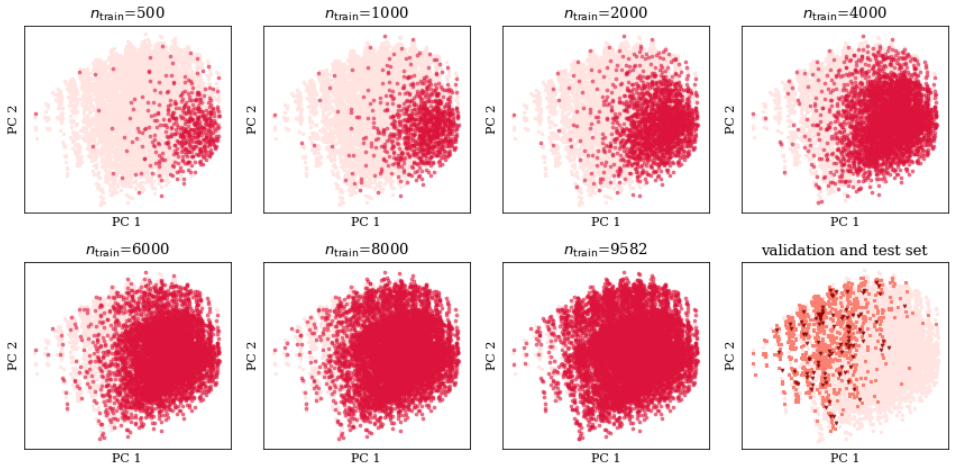
Supplementary Figure 15. kPCA plot for the intensive kernel and an intensive FPS split. The individual subplots show the distribution of training configurations with different training set sizes. The bottom right panel shows the distribution of validation (triangles) and test set (squares).



Supplementary Figure 16. kPCA plot for the intensive kernel and an extensive FPS split showing the distribution of the training, validation and test set configurations (see Supplementary Figure 15).



Supplementary Figure 17. kPCA plot for the extensive kernel and an extensive FPS split showing the distribution of the training, validation and test set configurations (see Supplementary Figure 15).

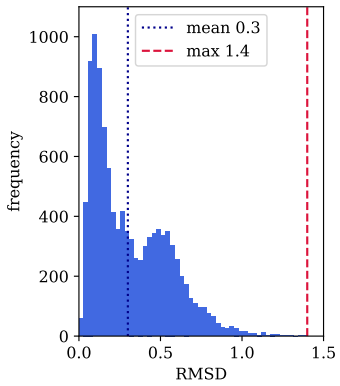


Supplementary Figure 18. kPCA plot for the extensive kernel and an intensive FPS split showing the distribution of the training, validation and test set configurations (see Supplementary Figure 15).

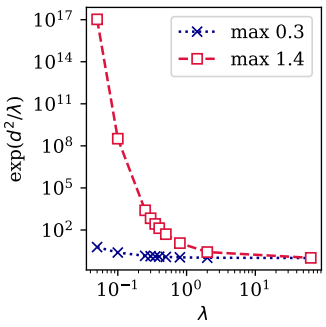
Supplementary Note 7: σ-Scaling

As discussed in the main text, for real applications using forcefield or semiempirical instead of DFT geometries is inevitable. The description of molecular geometries with UFF can be of varying quality for different molecules, which implies that there is not a constant level of noise on the reference data (see Supplementary Figure 19). To this end, Bartók et al.[10] suggested to weight training structures so that the ML model naturally assumes higher uncertainties for configurations that have poor geometries. In their work they quantify the difference between high and low level structures as the root mean square deviation (RMSD) d and scale the regularization parameter σ to be proportional to the factor $f = \exp(\frac{d^2}{\lambda})$. By this a new hyperparameter λ arises that has to be determined empirically. To estimate the range of reasonable parameters we plot the scaling factor f as a function of λ using the maximum and mean RMSD in the database (see Supplementary Figure 20). The plot shows a huge deviation between the scaling factors, especially for small λ. In this case structures with a large RMSD are scaled by 4-17 orders of magnitude and structures with an average RMSD by around 1 order of magnitude for the three lowest λ values.

The results of learning the atomization energies with and without σ-scaling for K_{ext} and K_{int} with both FPS splits are shown in Supplementary Figure 21. We found that σ-scaling does not effect the prediction of AEs using the intensive FPS split. For every point in the learning curve the RMSE for validation and test set remains the same and the λ values assume one of the highest values resulting in a scaling factor of 1.



Supplementary Figure 19. Histogram of RMSD values. Average (dotted) and maximum RMSD (dashed) are indicated.

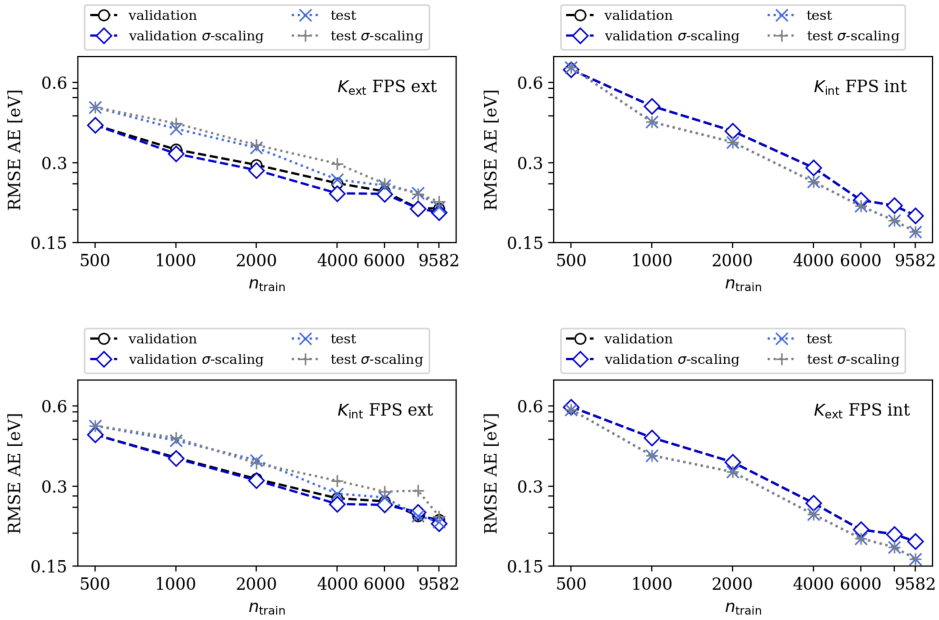


Supplementary Figure 20. Illustrative scaling values for the diagonal elements using the maximum RMSD $d = 1.4\,\text{Å}$ and average RMSD $d = 0.3\,\text{Å}$. The labels represent the used λ values in the grid search.

In contrast, the results change somewhat for the predictions using the extensive FPS split (left subplots in Supplementary Figure 21). In these cases, σ-scaling lowers the error of the validation set, but increases the RMSE in the test set for both the extensive and the intensive kernel and thus leads to some degree of over-fitting.

To conclude, an improvement of the predictions for AE using the RMSD of UFF and DFT training configurations to scale the regularization parameter was not successful. This is likely due to the different and poor quality of UFF geometries of open-shell structures.

In this work the RMSD values are calculated with the code `rmsd` obtained from GitHub.[24,25] Since the molecules for the UFF and DFT geometry optimization are created from the same smile string, no reordering was applied.



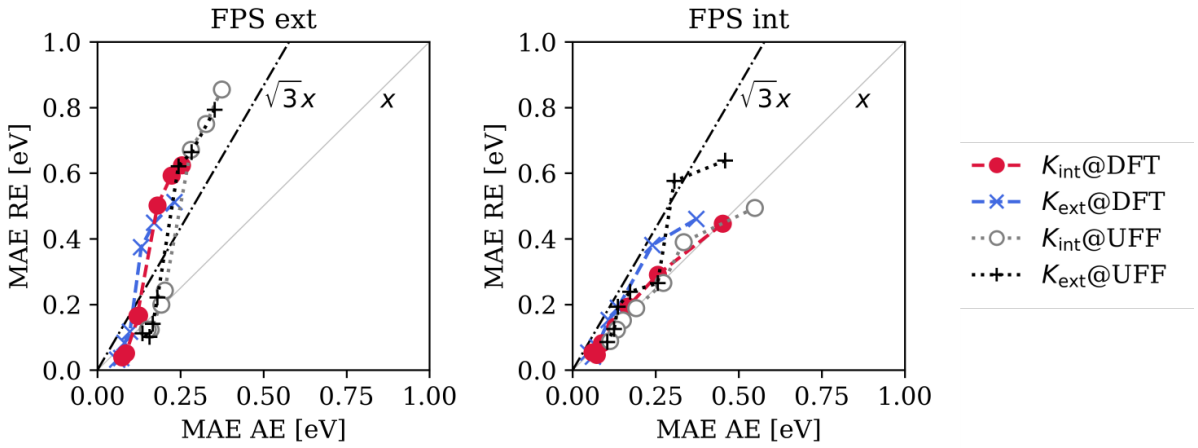
Supplementary Figure 21. Learning curves of AE predictions for validation and test structures with and without σ-scaling using the extensive and intensive kernels with both FPS splits. The RMSE is displayed because the hyperparameter are selected according to the minimum RMSE of the validation set.

Supplementary Note 8: Learning RE with UFF Geometries

Supplementary Figure 22 shows the correlation plots between the predicted atomization energies and reaction energies for DFT and UFF geometries. Comparable to the learning of AE, trends for RE with UFF geometries are similar to those with DFT, but with an higher MAE.

Supplementary Note 9: Training Set Selection with Random Sampling

In this section we show the performance for the prediction of AE and RE using random sampling for training set selection in contrast to the farthest point sampling used in the main manuscript. To this end we generated a randomly chosen sequence of up to 9582 training, 100 validation (for hyperparameter optimization) and 1030 test configurations. This split is applied to the predictions of atomization energies and corresponding reaction energies of Rad-6-RE using the extensive and the intensive kernels. kPCA plots illustrating the random



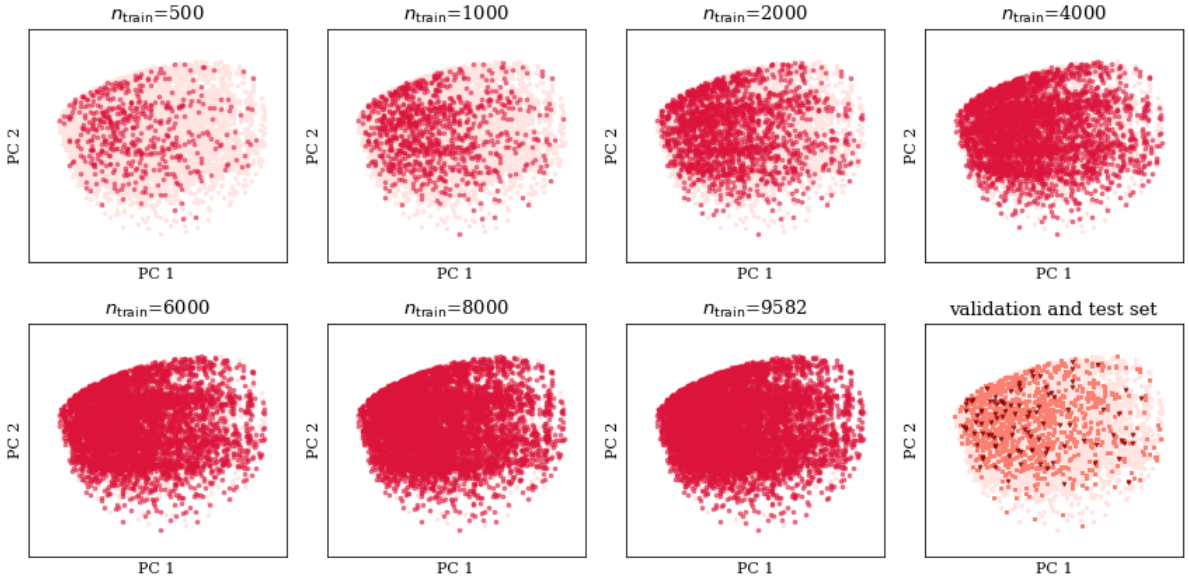
Supplementary Figure 22. Mean absolute errors (MAEs) for AE and RE predictions using DFT (dashed lines) and UFF (dotted lines) geometries and the extensive and intensive kernels described in the manuscript. Multiple points for each model represent the different training set sizes shown in Supplementary Figure 12.

sets are shown in Supplementary Figures 23 and 24.

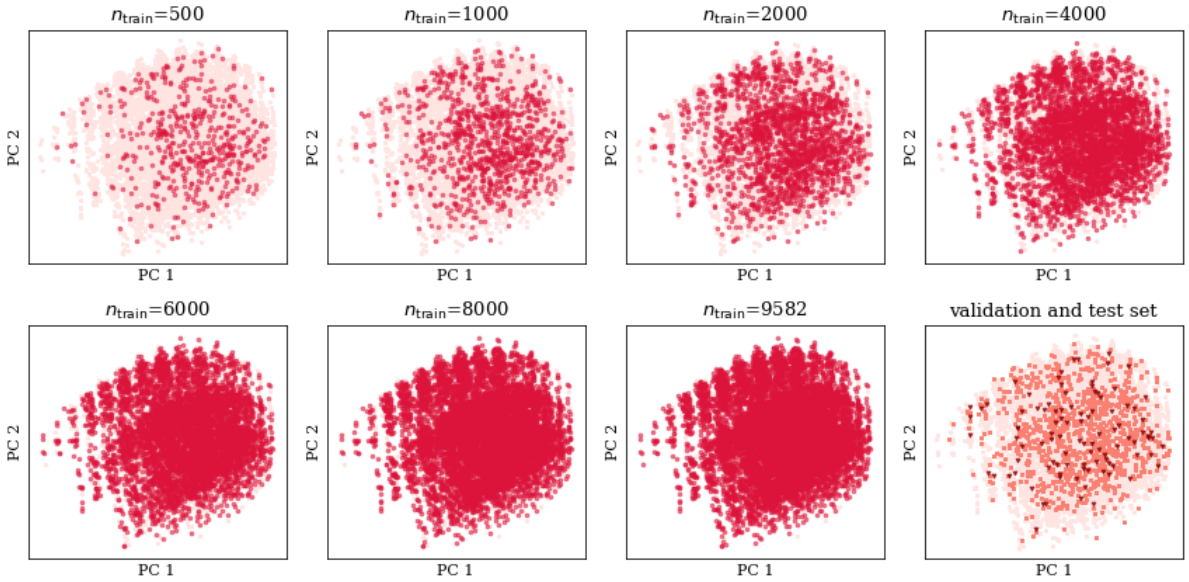
Hyperparameter search: The hyperparameter search was performed as described in Supplementary Note 3 (see also Supplementary Figures 25 and 27). However, an exception was made in the case of the intensive kernel. Here, only 99 molecules were used in the validation set to determine the regularisation parameter σ. This is because large errors for the carbon dimer lead to a poor choice of σ in this case (i.e. the models were severely underfitted). This illustrates the dangers of pure random sampling: C_2 has low similarity with all other molecules in the dataset and should therefore be included in the training set (see also Supplementary Figure 26).

Learning curves: The learning curves are analogous to Supplementary Figures 8-11 but use training sets from random sampling.

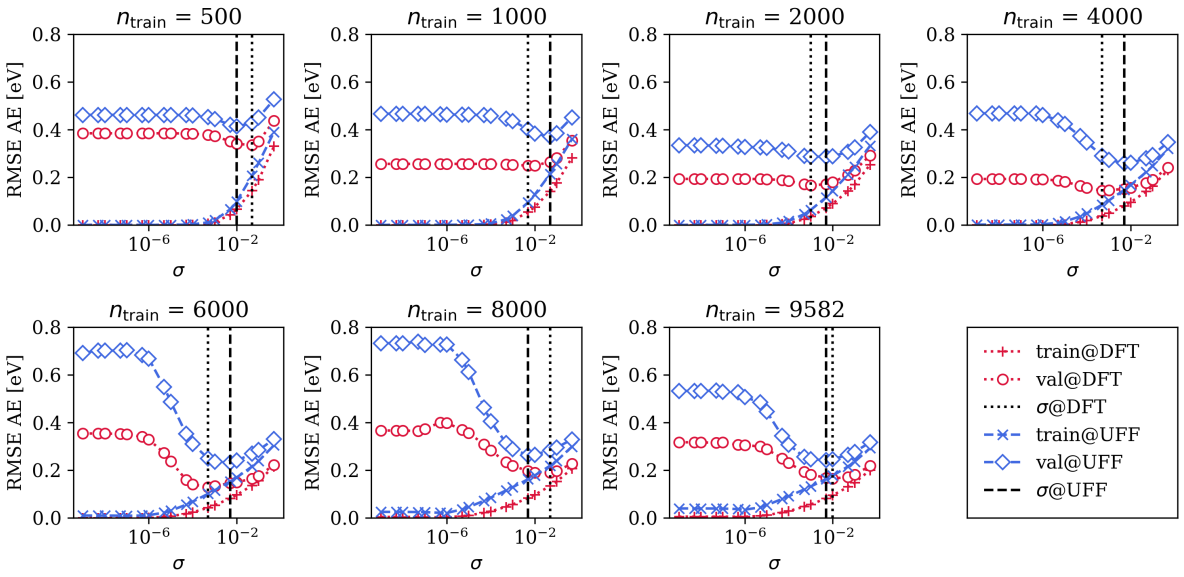
Learning atomization energies: Supplementary Figure 30 shows the results of the atomization energy predictions for random sampling (right subplot) together with both FPS splits for the intensive and extensive kernels. The general trends with respect to kernel selection and the effect of UFF vs. DFT geometries are the same in all cases. However, the prediction errors for large training sets are somewhat larger in the case of random sampling, though it is competitive for small training sets.



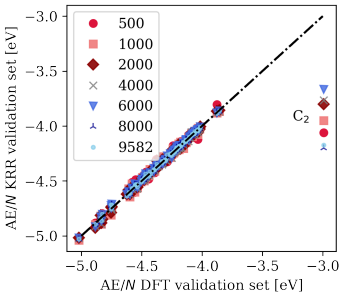
Supplementary Figure 23. kPCA plot for the intensive kernel and a random training set sampling. The individual subplots show the distribution of training configurations with different training set sizes. The bottom right panel shows the distribution of validation (triangles) and test set (squares).



Supplementary Figure 24. Same plot as Supplementary Figure 23 but for the extensive kernel with random training set selection.

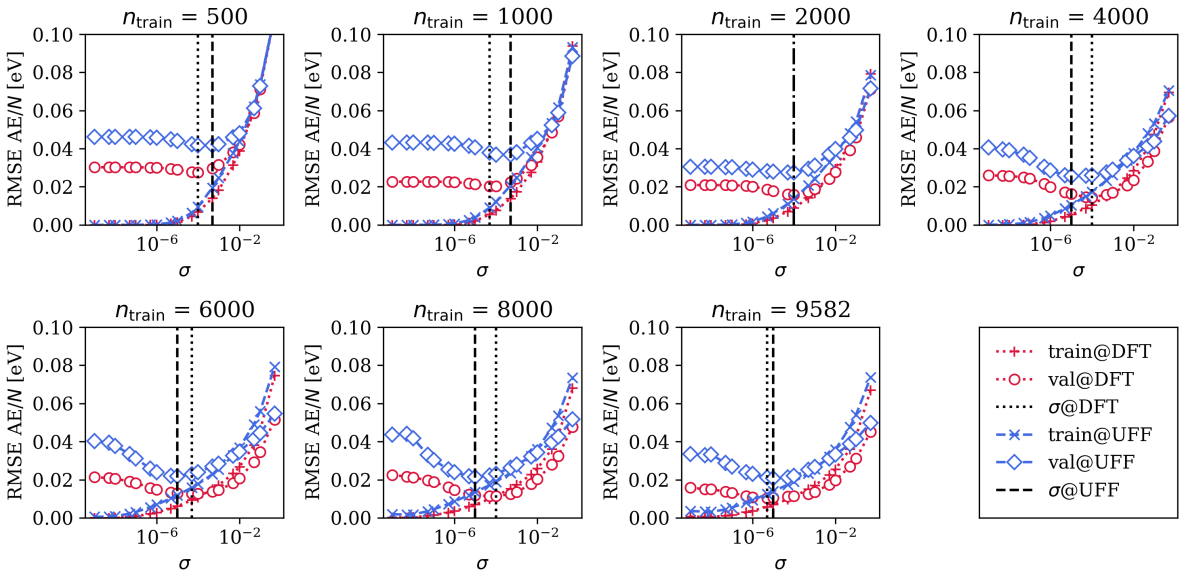


Supplementary Figure 25. Hyperparameter search for K_{ext} and random sampling.

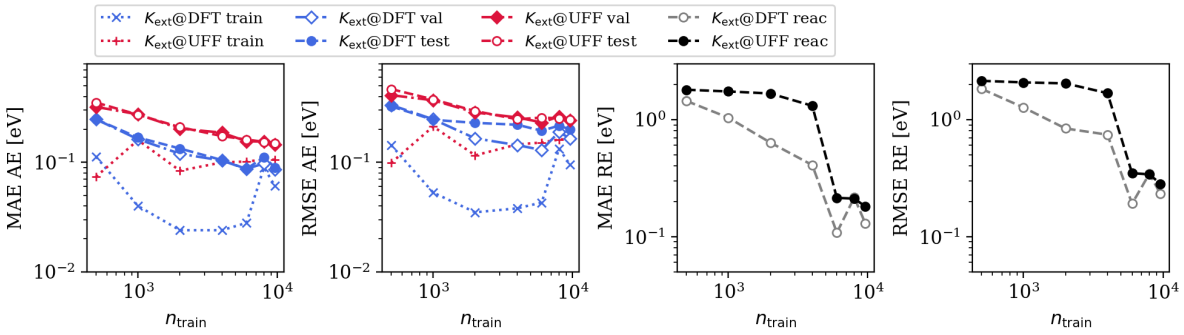


Supplementary Figure 26. Correlation plot of DFT calculated and predicted AE/N for the validation set with the intensive kernel using random sampling and different training set sizes.

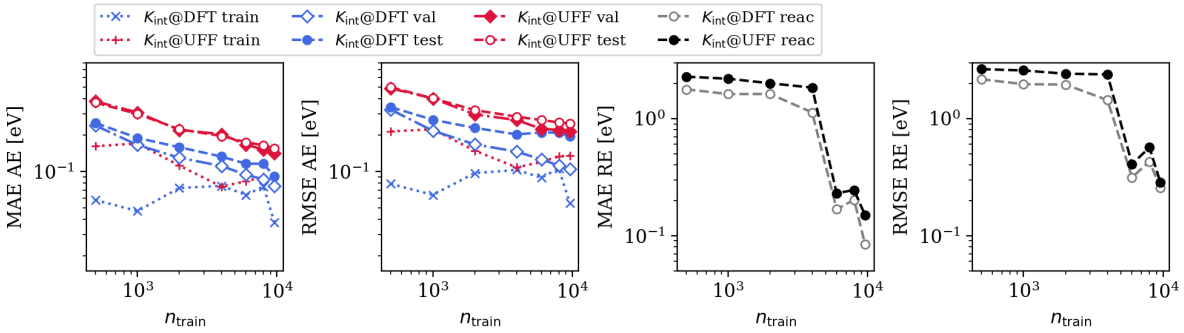
Learning reaction energies: In this section we compare the results of random training set selection with farthest point sampling for the prediction of reaction energies in the Rad-6-RE network (see Supplementary Figure 31). We see that for small training set sizes, random sampling performs drastically worse for the predictions of reaction energies. Similar to what is observed for the extensive FPS split, this is attributed to large errors for essential 'hub' molecules, which are absent from the training set. This is only mitigated for the larger training sets, which approach the FPS sets (though still displaying larger MAEs).



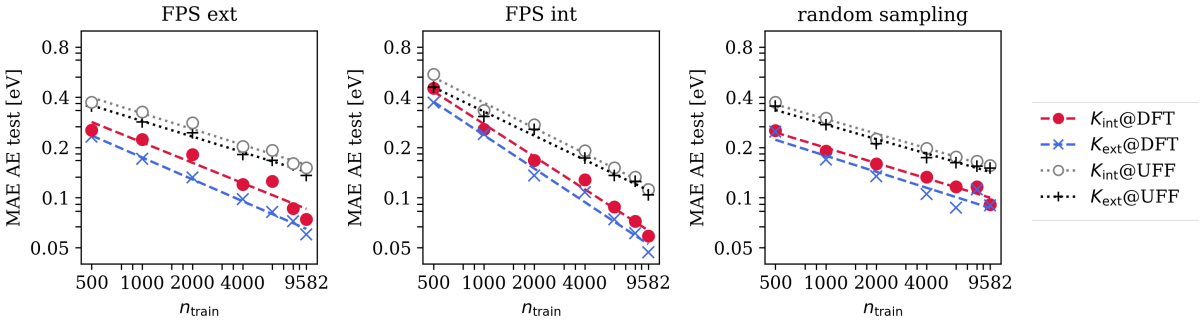
Supplementary Figure 27. Hyperparameter search for K_{int} and random sampling.



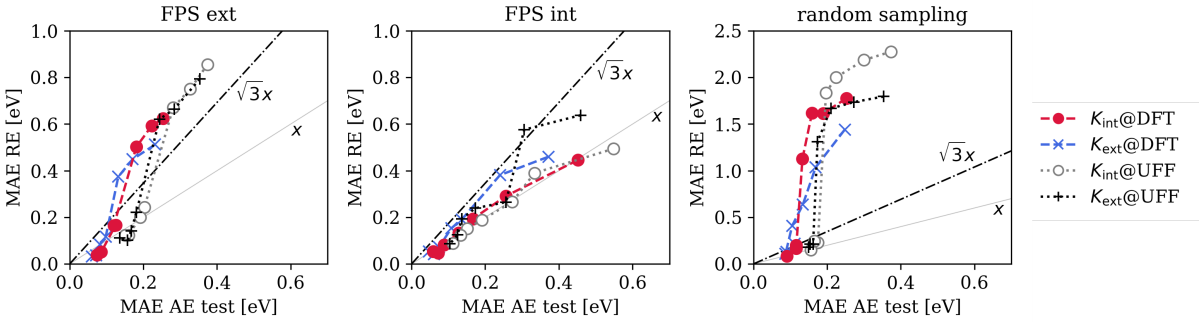
Supplementary Figure 28. Learning curves for K_{ext} and random sampling.



Supplementary Figure 29. Learning curves for K_{int} and random sampling.



Supplementary Figure 30. Comparison of learning curves for atomization energy (AE) predictions using extensive and intensive kernels for both DFT and UFF geometries. The three subplots show the results for the entensive and intensive FPS splits as well as for random sampling.

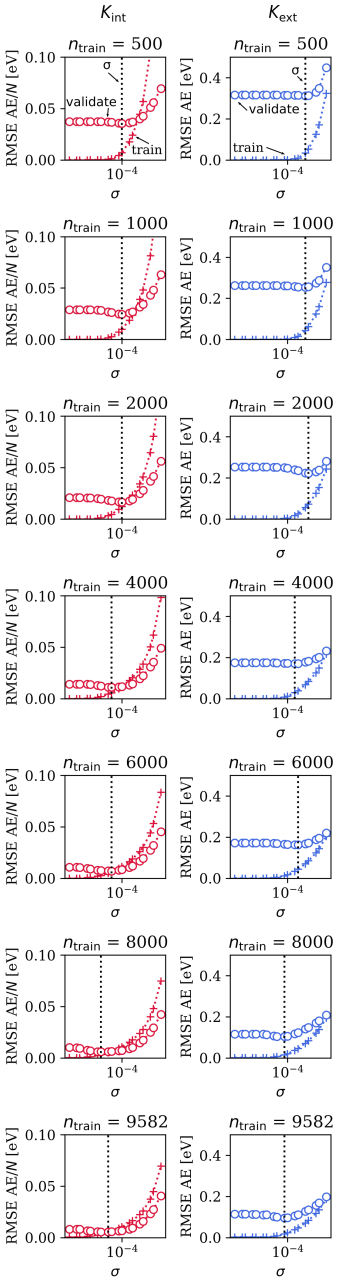


Supplementary Figure 31. Mean absolute errors (MAEs) for AE and RE predictions using DFT (dashed lines) and UFF (dotted lines) geometries and the extensive and intensive kernels for both FPS splits and random sampling. Multiple points for each model represent the different training set sizes shown in Supplementary Figure 30.

Supplementary Note 10: Comparison Rad-6 and Rad-6-BS

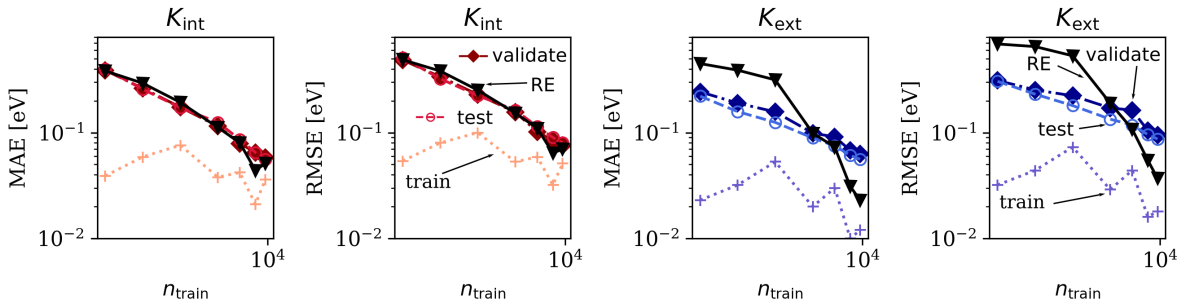
As discussed in the manuscript, the choice of reference spin-states taken for Rad-6 is somewhat arbitrary and may not be ideal for every application. Nonetheless, we expect the ML methodology developed herein to be equally applicable to reference data with different choices in spin-states. In this light, it is instructive compare the results from the main manuscript with models trained on the Rad-6-BS database (for computational details see Supplementary Note 1.)

The corresponding hyperparameter searches, learning curves and final results are shown



Supplementary Figure 32. Hyperparameter search for ML models of the Rad-6-BS database. The panels show the hyperparameter surfaces for the intensive (left) and extensive (right) kernel.

in Supplementary Figures 32-33 . These results are obtained with K_{int} and K_{ext} and the corresponding FPS split. As expected the differences in the reference methods between Rad-6 and Rad-6-BS do not significantly affect the performance of the ML models for AE and RE prediction.



Supplementary Figure 33. Learning curves for Rad-6-BS predictions showing the MAE and RMSE for the intensive and extensive kernel. Dotted positive signs are the AE errors of the training sets, solid diamonds are the AE errors of the validation sets, dashed circles are the AE errors of the test sets and solid triangles are the errors of the reaction energies.

Supplementary Note 11: Microkinetic Simulation

In the main text, we explore a realistic reaction network consisting of 21,392 reactions using an approximate microkinetic simulation. This network contains bond-breaking, transfer and rearrangement reactions of the general form:

$$A + B \rightleftharpoons C + D,$$

where molecules B and/or D can be 'empty' placeholders for bond-breaking and rearrangement reactions.[2]

The kinetics of this reaction network are governed by differential equations of the form:

$$\frac{d\theta_A}{dt} = -\sum_{B,CD} 2^{\delta_{AB}} \theta_A \theta_B k_{AB}^{CD} + \sum_{CD,B} 2^{\delta_{AB}} \theta_C \theta_D k_{CD}^{AB},$$

where θ_A is the concentration of molecule A, k_{AB}^{CD} is the rate constant for the reaction $A + B \rightarrow C + D$. Note that the first sum is over all elementary reactions that consume A, and the second sum is over the corresponding reverse reactions, where A is formed.

The term $\theta_A \theta_B k_{AB}^{CD}$ corresponds to the current rate of a given reaction, r_{AB}^{CD}. In other words, the rate depends on the concentration of the educts and the rate constant k_{AB}^{CD}, which is in turn proportional to the reaction energy and the activation energy. As mentioned in the main text, all activation energies are assumed to be identical. We can then compute the rate constants from transition state theory *via*:

$$k_{AB}^{CD} = e^{\frac{-\Delta E}{k_B T}}$$

Here, the energy difference ΔE is the reaction energy plus the activation energy for an endothermic reaction and the activation energy for an exothermic reaction. Under these circumstances, the actual value of the activation energy is not important (it is chosen to be 0.3 eV), and only changes the arbitrary time unit of the simulation. Similarly, we choose a constant pre-exponential factor of 1 for all reactions.

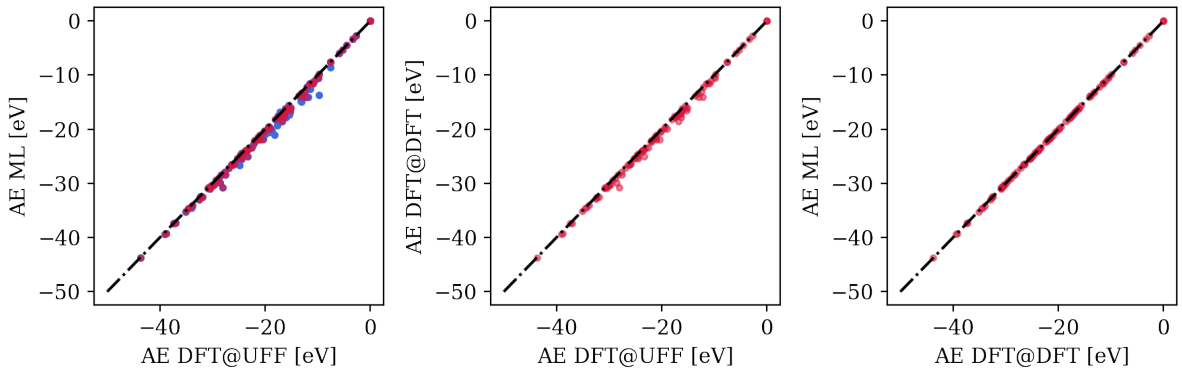
The simulation is initialized with equal concentrations of CH_4 and O_2, all other concentrations set to 0. At the beginning of the simulation, all rates are thus also 0, except for reactions involving CH_4 and O_2. We then propagate the differential equations specified above using a third-order Runge-Kutta integrator.[26] As the concentrations are updated, more rates become larger than zero. The subgraphs shown in the main manuscript show all reactions with non-zero rates at a given simulation time.

Supplementary Note 12: Validation of Out-Of-Sample Predictions

As discussed in the main manuscript, the reaction network used in the microkinetic analysis contains several systems that are not included in Rad-6, and thus represent a true out-of-sample application of the ML model. To evaluate the quality of these predictions, DFT calculations were preformed on these out-of-sample systems. Unfortunately, these systems are missing from Rad-6 because they either decomposed upon geometry relaxation or had SCF convergence issues in the original high-throughput simulations for the database. We were, however, able to obtain single-point DFT energies on frozen UFF geometries (DFT@UFF, same computational settings as for Rad-6) for all but one of these systems.

In Supplementary Figure 34, correlation plots for DFT@UFF, DFT@DFT and ML predicted AEs are shown (with the out-of-sample systems highlighted in blue). As expected, the ML and DFT@DFT values display an excellent correlation. Meanwhile, both of these approaches consistently predict more negative AEs than the DFT@UFF approach, since the latter is missing geometry relaxation effects. Importantly, this is also the case for the out-of-sample predictions, meaning that the ML model can be used to estimate relaxation effects even when DFT relaxations are not available.

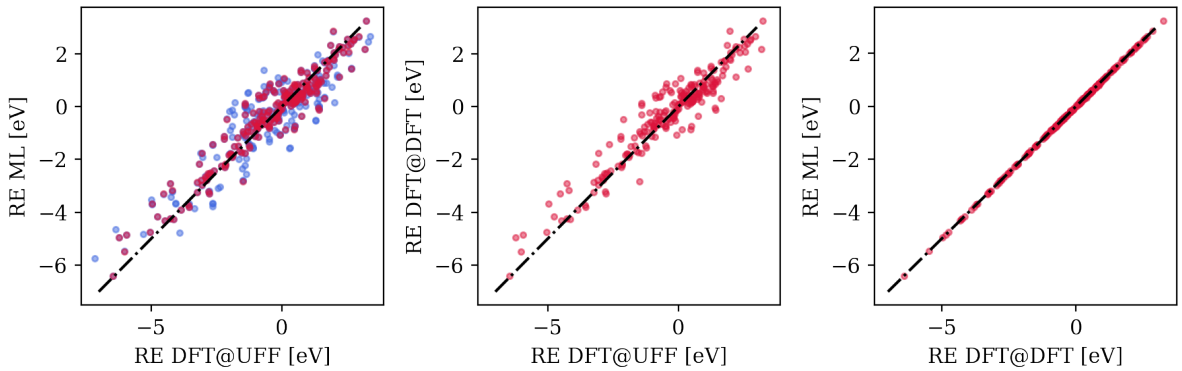
Overall, there is also a good correlation between the DFT@UFF values and the ML predictions, with R^2 =0.994. To quantify the magnitude of geometry relaxation effects, we calculate the mean error (ME) between DFT@UFF and ML, in addition to the MAE. We



Supplementary Figure 34. Correlation plots for predicted atomization energies. Left: ML model (used in main text) vs. single point DFT calculations at UFF geometries (DFT@UFF). Middle: Optimized DFT calculations (DFT@DFT) vs. DFT@UFF. Right: ML model vs. DFT@DFT. Note that the left panel contains the additional out-of-sample data points (highlighted in blue, see text for details).

find that the ME and MAE are nearly identical (ca. 0.6 eV, see Supplementary Table 2), confirming the systematic nature of the deviation. For comparison, the corresponding DFT@DFT values are also shown, again with identical ME and MAE. Note that the deviations relative to DFT@UFF are not identical for ML and DFT@DFT because the ML comparison includes more systems (the out-of-sample set). Taken as a whole, these observations provide a strong indication that our ML model predicts reasonable AEs for the out-of-sample molecules in the network.

This data is also used to verify the reaction energies that go into the microkinetic simulations, as shown in Supplementary Figure 35. Again, we find a good correlation between our ML model and the DFT@UFF calculations, with some scatter. Importantly, similar correlation and scatter are observed when comparing DFT@DFT and DFT@UFF, confirming the high quality of the ML predictions.



Supplementary Figure 35. Correlation plots for reaction energies. Labels are analogous to Supplementary Figure 34. Shown reaction energies are from the reduced network at t=128 (see manuscript for details).

Supplementary Table 2. Summary of statistics (MAE, ME and R^2) pertaining to the plots in Supplementary Figures 34 and 35.

	MAE AE [eV]	ME AE [eV]	R^2	N
DFT@UFF - ML	0.606	0.599	0.994	130
DFT@UFF - DFT@DFT	0.414	0.414	0.997	101
DFT@DFT - ML	0.022	0.0005	1.000	101
	MAE RE [eV]	ME RE [eV]	R^2	N
DFT@UFF - ML	0.572	-0.044	0.840	365
DFT@UFF - DFT@DFT	0.420	-0.074	0.910	225
DFT@DFT - ML	0.009	0.0004	1.000	225

SUPPLEMENTARY REFERENCES

[1]Weininger, D. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31–36.

[2]Margraf, J. T.; Reuter, K. *ACS Omega* **2019**, *4*, 3370–3379.

[3]RDKit: Open-source cheminformatics. `http://www.rdkit.org`.

[4]Rappe, A. K.; Casewit, C. J.; Colwell, K. S.; Goddard, W. A.; Skiff, W. M. *J. Am. Chem. Soc.* **1992**, *114*, 10024–10035.

[5]Blum, V.; Gehrke, R.; Hanke, F.; Havu, P.; Havu, V.; Ren, X.; Reuter, K.; Scheffler, M.

Comput. Phys. Commun. **2009**, *180*, 2175 – 2196.

[6]Zhang, I. Y.; Ren, X.; Rinke, P.; Blum, V.; Scheffler, M. *New J. Phys.* **2013**, *15*, 123033.

[7]Adamo, C.; Barone, V. *J. Chem. Phys.* **1999**, *110*, 6158–6170.

[8]Tkatchenko, A.; DiStasio, R. A.; Car, R.; Scheffler, M. *Phys. Rev. Lett.* **2012**, *108*, 236402.

[9]Rupp, M. *Int. J. Quantum Chem.* **2015**, *115*, 1058–1073.

[10]Bartók, A. P.; De, S.; Poelking, C.; Bernstein, N.; Kermode, J. R.; Csányi, G.; Ceriotti, M. *Sci. Adv.* **2017**, *3*, e1701816.

[11]Neese, F. *WIREs Comput. Mol. Sci.* **2018**, *8*, e1327.

[12]Zhang, Y.; Yang, W. *Phys. Rev. Lett.* **1998**, *80*, 890.

[13]Vaucher, A. C.; Reiher, M. *J. Chem. Theory Comput.* **2017**, *13*, 1219–1228.

[14]Deringer, V. L.; Csányi, G. *Phys. Rev. B* **2017**, *95*, 094203.

[15]Szlachta, W. J.; Bartók, A. P.; Csányi, G. *Phys. Rev. B* **2014**, *90*, 104108.

[16]Cliffe, M. J.; Bartók, A. P.; Kerber, R. N.; Grey, C. P.; Csányi, G.; Goodwin, A. L. *Phys. Rev. B* **2017**, *95*, 224108.

[17]Bartók, A. P.; Kondor, R.; Csányi, G. *Phys. Rev. B* **2013**, *87*, 184115.

[18]De, S.; Bartók, A. P.; Csányi, G.; Ceriotti, M. *Phys. Chem. Chem. Phys.* **2016**, *18*, 13754–13769.

[19]`https://github.com/libAtoms/QUIP`.

[20]`https://github.com/simonwengert/mltools.git`.

[21]Ceriotti, M.; Willatt, M. J.; Csányi, G. Machine-learning of atomic-scale properties based on physical principles. `https://arxiv.org/abs/2001.11696`.

[22]Rasmussen, C.; Williams, C. *Gaussian Processes for Machine Learning*; MIT Press, 2006.

[23]Schölkopf, B.; Smola, A.; Müller, K.-R. *Neural Comput.* **1998**, *10*, 1299–1319.

[24]Kabsch, W. *Acta Cryst. A* **1976**, *32*, 922–923.

[25]`https://github.com/charnley/rmsd`.

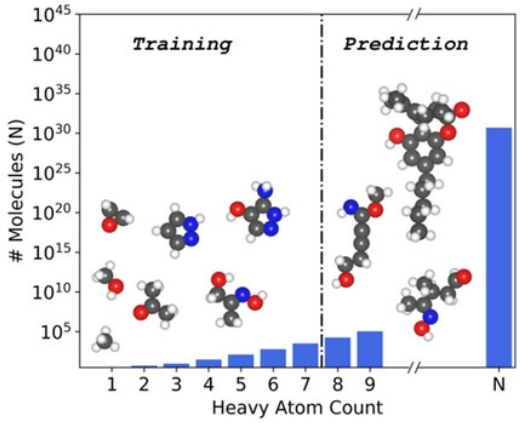
[26]Bogacki, P.; Shampine, L. *Applied Mathematics Letters* **1989**, *2*, 321 – 325.

Paper # 2

Size-Extensive Molecular Machine Learning with Global Representations
Hyunwook Jung*, Sina Stocker*, Christian Kunkel, Harald Oberhofer, Byungchan Han, Karsten Reuter and Johannes T. Margraf
ChemSystemsChem 2, e1900052 (2020).
DOI: https://doi.org/10.1002/syst.201900052

Abstract

A sizeable difference: The machine-learning (ML) based exploration of chemical space requires models that can appropriately handle molecules of different sizes. To achieve this, the size-extensivity of ML models should be enforced. In this paper conditions for extensive ML models are discussed, and extensive models are shown to effectively extrapolate to large molecules, when trained on small ones.



*These authors contributed equally

Chemistry Europe
European Chemical Societies Publishing

Special Collection

Size-Extensive Molecular Machine Learning with Global Representations**

Hyunwook Jung[+],[a, b] Sina Stocker[+],[a] Christian Kunkel,[a] Harald Oberhofer,[a] Byungchan Han,[b] Karsten Reuter,[a] and Johannes T. Margraf*[a]

Machine learning (ML) models are increasingly used in combination with electronic structure calculations to predict molecular properties at a much lower computational cost in high-throughput settings. Such ML models require representations that encode the molecular structure, which are generally designed to respect the symmetries and invariances of the target property. However, size-extensivity is usually not guaranteed for so-called global representations. In this contribution, we show how extensivity can be built into global ML models using, *e.g.*, the Many-Body Tensor Representation. Properties of extensive and non-extensive models for the atomization energy are systematically explored by training on small molecules and testing on small, medium and large molecules. Our results show that non-extensive models are only useful in the size-range of their training set, whereas extensive models provide reasonable predictions across large size differences. Remaining sources of error for extensive models are discussed.

1. Introduction

In recent years, machine-learning (ML) methods are increasingly applied to the prediction of molecular properties such as atomization and orbital energies, dipole moments and ionization potentials.[1–9] One of the main promises of ML in chemistry is that it allows surpassing the size and time scales accessible to accurate first-principles electronic structure calculations, *e.g.* based on density-functional theory (DFT). This is particularly relevant in a high-throughput setting, *e.g.* when a large chemical reaction network with many intermediates and transition states is to be explored, or a large chemical space is of interest.[10–13]

The wide range of ML methods that have emerged in this context raises the question which one should be used for a given application. Since the atomization energy (AE) has a long tradition as the foremost benchmark property to judge the accuracy of quantum chemical approximations,[14–16] it has also become one of the standard targets to illustrate the accuracy of novel ML methods.[1,3] The most straightforward way to construct a ML model for the AE is to use some vectorized representation v of the molecule . Constructing the ML model is then simply a regression task between v and the property of interest $y(v)$.[17] While any general linear or non-linear regression method (*e.g.* Kernel Ridge Regression, KRR or Artificial Neural Networks) can be used, the choice of the representation is critical. In particular, several physically motivated criteria such as translational, permutational, and rotational invariance and uniqueness should be fulfilled.[5,18]

The Coulomb matrix (CM) developed by Rupp et al.[4] was one of the earliest (global) molecular representations used to this end (see below for a specification of global in contrast to local representations). However, it suffers from two notable limitations, namely that the size of v depends on the number of atoms in the system and that permutational invariance can only be achieved through a canonical ordering of the vector elements.[2] This led to several subsequent improvements of the CM concept, such as the Bag-of-Bonds,[19] different histogram based methods[1] and the Many-Body Tensor Representation (MBTR).[6,18] These representations fix the main drawbacks of the CM and can thus be used to construct more accurate and data-efficient ML models of molecular properties, typically using KRR.

However, the combination of KRR with global representations still suffers from the problem that the resulting predictions are typically not size-extensive. This should in principle be a fundamental problem for predicting any extensive property like the AE. In practice, this issue can be and has been overlooked to some extent, as the databases that are hitherto typically used to test ML models (*e.g.* QM9)[20] do not contain large size differences. For example, ca. 97% of the molecules in QM9 contain 8 or 9 heavy atoms. Consequently, an approximate size-extensivity of the model can be learned by simply including all small systems in the training set explicitly.[17] However, this only obscures the fundamental problem, and such a model will fail when applied to significantly larger molecules. Similarly, the

[a] H. Jung,[+] S. Stocker,[+] C. Kunkel, PD Dr. H. Oberhofer, Prof. Dr. K. Reuter, Dr. J. T. Margraf
Chair for Theoretical Chemistry and Catalysis Research Center, Technische Universität München, Lichtenbergstraße 4, D-85747 Garching, Germany
E-mail: johannes.margraf@ch.tum.de
[b] H. Jung,[+] Prof. Dr. B. Han
Department of Chemical and Biomolecular Engineering, Yonsei University, Seoul 03722, Republic of Korea
[+] These authors contributed equally to the manuscript

[**] A previous version of this manuscript has been deposited on a preprint server (DOI: 10.26434/chemrxiv.10002020).

Supporting information for this article is available on the WWW under https://doi.org/10.1002/syst.201900052

Special Collection An invited contribution to a Special Collection on the Computational Chemistry of Complex Systems

description of chemical reactions (where a large molecule can decompose into smaller fragments) cannot be consistently achieved when the predicted energies are non-extensive.[21]

The goal of the present paper is to address the size-extensivity of ML models that use a global representation of the molecular structure, using KRR models with the MBTR of Huo and Rupp as an illustrative example.[18] We will discuss how extensive ML models can be constructed with MBTR and compare them with the conventional, non-extensive formulation. Importantly, the performance of the models is compared across different size-ranges both within the QM9 database and between databases going up to molecules with more than 80 heavy atoms.[11,22]

2. Theory

Kernel Ridge Regression: In KRR, the target property $y(v)$ (*i.e.* here the AE) of an unknown molecule with the representation v is calculated via:

$$y(v) = \sum_i w_i K(v, v_i) \tag{1}$$

where v_i are the representations of training data points and w_i are regression weights. Here, we introduced the kernel function $K(v, v')$, which provides a similarity measure between two representations v and v'. A common choice for $K(v, v')$ is the Gaussian kernel:

$$K(v, v') = \exp\left(-\frac{\|v - v'\|_2^2}{2\sigma^2}\right). \tag{2}$$

Here, σ is the kernel length scale, a hyperparameter that governs how prone the kernel is to classify systems as similar. Specifically, a large value of σ will indicate some degree of similarity between most inputs, whereas a small value will only find similarities for systems that are very close in feature space. Below, we also use the linear kernel, which simply consists of the dot-product of v and v'.

The optimal (in a least-squares sense) set of weights ω can be obtained via the expression:

$$w = (\mathbf{K} + \lambda \mathbf{I})^{-1}\mathbf{y}, \tag{3}$$

where $\mathbf{K}$ is the kernel matrix of the training set (with $K_{ij} = K(v_i, v_j)$), λ is a regularization parameter and $\mathbf{I}$ is the identity matrix. λ is another hyperparameter of the model, which represents the uncertainty of the observations.

Training a KRR model is thus a simple linear algebra operation. Obviously, the performance of the model critically depends on the choice of representation and kernel function. In analogy to the common notation of Functional/Basis-Set in DFT, this choice is designated as Representation/Kernel in the following.

Many-Body Tensor Representation: Herein, we use the MBTR of Huo and Rupp as a prototypical global representation of molecular structure.[18] Simply put, the MBTR provides a measure of how often characteristic geometric features (corresponding to different orders of a many-body expansion) occur. Canonically, these features are atom counts (1-body), inverse interatomic distances (2-body), angles (3-body), dihedrals (4-body), etc. For each body-order and element combination, a broadened distribution function of these features is constructed as a sum of Gaussians, as shown in Figure 1 for the 2-body terms in water. These Gaussians are additionally scaled by a distance-dependent weighting function, which introduces a characteristic length-scale to the representation. Beyond this length-scale atoms or molecules are effectively non-interacting.

For a given body order k and N_{species} chemical species there are in principle $N_{\text{max}} = N_{\text{species}}^k$ such distribution functions. Although some combinations can be excluded by symmetry (*i.e.* C–H is equal to H–C), this means that the size of the MBTR vector quickly explodes with the body order. In practice, the MBTR is therefore usually limited to the lowest order terms, *i.e.* including up to 2- or 3-body contributions. The final MBTR vector is obtained by concatenating the discretized feature distribution functions $v_{k,i}$:

$$v^{MBTR} = v_{1,1} \oplus v_{1,2} \oplus \ldots \oplus v_{k_{max},N_{max}} \tag{4}$$

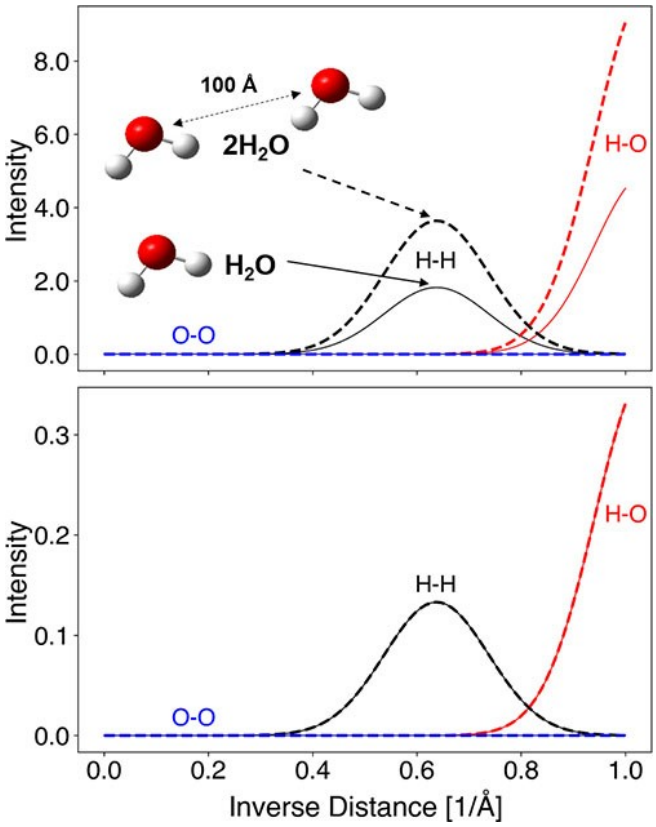


Figure 1. Sample illustration of 2-body MBTR output of a single water molecule (solid) and two distant water molecules (dashed). Interatomic interactions include: H–H (black), H–O (red), and O–O (blue). (Top) MBTR (Bottom) iMBTR

In this original formulation, the representation thus contains absolute counts of the occurrence of a given feature. In contrast, below, we also consider a *normalized* version of the MBTR, where each distribution function is normalized according to its l^2-norm:

$$v^{\text{iMBTR}} = \frac{1}{\|v_{1,1}\|} \cdot v_{1,1} \oplus \frac{1}{\|v_{1,2}\|} \cdot v_{1,2} \oplus \ldots \oplus \frac{1}{\|v_{k_{\max},N_{\max}}\|} \cdot v_{k_{\max},N_{\max}} \quad (5)$$

For clarity, this normalized MBTR version is designated as iMBTR (for *intensive*).

Size-Extensivity: According to eq. 1, the target property (here the AE) is predicted as a linear combination of kernel functions. Consequently, it is advantageous if the kernel can be constructed in such a way that it adheres to conditions known to be fulfilled by the target property. For example, the AE is invariant to translations and rotations of a molecule. Consequently, MBTR-based kernels are constructed to satisfy these same invariances.

A less commonly imposed condition relates to the extensive or intensive nature of the target property. As with the invariances, the kernel should ideally reflect the extensivity or intensivity of the property of interest. Specifically, for two non-interacting molecules A,

$$K(A, 2A) = 2 \times K(A, A), \quad (6)$$

for an extensive property (such as the AE) and

$$K(A, 2A) = K(A, A), \quad (7)$$

for an intensive property (such as the ionization potential).

Unfortunately, the original MBTR/Gaussian kernel is neither intensive nor extensive. While the distribution functions that make up the representations for A and $2A$ have identical shapes, the amplitude of each peak is twice as large for $2A$ (see Figure 1, top). Since the norm of the difference between MBTR vectors enters the Gaussian kernel, it will evaluate to approximately zero (depending on the lengthscale σ). In contrast, the combination iMBTR/Gaussian leads to an *intensive* kernel. This is because the iMBTR for an arbitrary number of non-interacting molecules becomes identical to the single molecule case due to its normalization (see Figure 1, bottom). Finally, the combination MBTR/linear leads to an *extensive* kernel. This can easily be verified by considering that each element in the MBTR of 2A differs from the MBTR of A by a factor of two.

From this perspective, the MBTR/linear kernel appears to be the most appropriate choice for learning AEs. However (as the name implies) KRR with the linear kernel is simply linear regression. As the main advantage of KRR is the introduction of non-linearity (*e.g.* via the Gaussian kernel), this is not ideal.

Fortunately, we can resort to a simple trick to obtain an extensive non-linear KRR model. Specifically, an iMBTR/Gaussian model can be trained to predict the atomization energy per atom (AE/N), which is an intensive quantity. Indeed, it has already been suggested in the context of electronic structure methods that AE/N may actually be a more appropriate target for fitting and benchmarking.[21,23]

Note that this *intensive atomization energy* should not be interpreted as a *local* atomic energy (see below). Instead it can be understood as a generalization of the concept of cohesive energy for extended crystals to finite systems.[23] In Figure 2, AE/N is plotted for linear hydrocarbons (*i.e.*, alkanes, alkenes, and alkynes) of different sizes. All three curves converge to a constant value (the cohesive energy of the corresponding 1D crystal) for large systems and display a smooth dependence on the number of atoms for smaller systems. To predict the AE with the iMBTR/Gaussian model, we thus train on AE/N and subsequently simply multiply the prediction by the number of atoms. For comparison, the original MBTR/Gaussian and MBTR/linear models are trained on the AE, as usual.

Global and Local representations: So far, we have focused on the general case of a *global* representation v, which encodes the entire structure of the molecule/system with the property $y(v)$. A major advantage of global ML models is that the assumed relationship between structure and property mirrors the fact that any property can in principle be computed from the Schrödinger equation.[24,25] Meanwhile, a significant drawback is that the cost of computing global representations does not scale linearly with the size of the system. This inhibits the use of global representation as universal descriptors applicable to proteins or solids. Fortunately, this is not problematic for molecular systems with tens to hundreds of atoms. A second, more critical aspect is that global representations are not automatically size-extensive, as discussed in the previous section.

In contrast to this, a variety of *local* ML models have been developed that guarantee size-extensivity and linear scaling.[26–28] In the tradition of empirical interatomic potentials, these models approximate the total property (here the AE) as a sum of local (*e.g.* atomic) contributions:

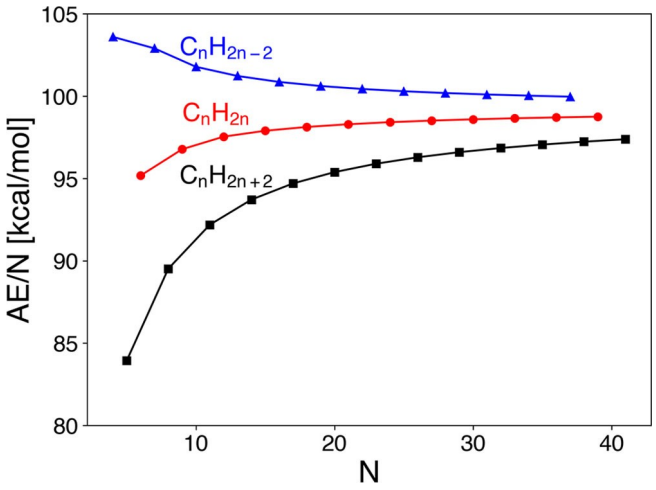


Figure 2. Atomization energy per atom for linear alkanes (C_nH_{2n+2}), alkenes (C_nH_{2n}) and alkynes (C_nH_{2n-2}) from C_1 to C_{13}.

$$y \approx \sum_{\text{atom}} y_{\text{atom}} \tag{8}$$

Here, the local properties y_{atom} (*e.g.* atomic energies in the case of AE) only depend on the immediate chemical environment of the atom. Importantly, these local energies result from an optimal decomposition of the total property and are not necessarily physically meaningful.

While the expression in Eqs. 8 is manifestly extensive, it also generally introduces an approximation to the model. For instance, in the case of total energies or AEs it effectively neglects any long-range interatomic interactions. Furthermore, the local properties (like a local energy) might not necessarily be quantum mechanical observables. In practice, the severity of this approximation is property and material dependent. For example, in many cases excellent interatomic potentials based on Eqs. 8 have been obtained.[29,30]

For kernel-based regression, there is an interesting connection between global and local representations, as there are several ways to convert local to global kernels. For example, as noted by Bártok and coworkers, a ML potential based on the local SOAP representation is equivalent to a global model using the *averaged* kernel:[3]

$$K(A,B) = \sum_{i \in A, j \in B} \frac{1}{N_A N_B} k(i,j), \tag{9}$$

where $K(A,B)$ is a global kernel comparing molecules A and B, and $k(i,j)$ is a local kernel comparing atoms i and j. Similarly, a sum of local kernels can also form a global kernel:[31]

$$K(A,B) = \sum_{i \in A, j \in B} k(i,j), \tag{10}$$

From the perspective taken in this paper, Eqs. 9 and 10 are recipes to construct global kernels from local representations, which conform to Eqs. 7 and 6, respectively. These kernels are special cases of the general case discussed herein, in the sense that local representations can be used to build extensive kernels, but not all extensive kernels must be built from local representations. Recently, Tamblyn and coworkers also suggested semi-local, extensive ML models based on deep neural networks.[32]

3. Methods

Datasets: In this paper, we use two reference databases of DFT AEs, namely the QM9 and OE62 sets.[20,22] The QM9 set includes over 134,000 drug-like organic molecules and is frequently used as a benchmark for ML studies.[1,3,6] The molecules in QM9 have a heavy atom count (HAC) of up to nine and are comprised of the elements H, C, O, N, and F. As alluded to above, most of these molecules (ca 97%) contain 8 or 9 heavy atoms. This leaves a total of 3993 molecules with a HAC = 1–7, which we will use for training.

The OE62 dataset originates from a high-throughput screening study for organic semiconductors by Schober et al. and has also been used for benchmarking different ML methods.[6,22] While somewhat smaller than QM9 (61,489 molecules) it is significantly more chemically diverse. For example, OE62 contains 16 different elements and much larger molecules, with up to 174 atoms (max. HAC = 92).

Predicting properties of the OE62 set is therefore a very hard task for ML models trained on the small molecules contained in QM9, but it should in principle be possible for a size-extensive model. However, this can only work if both datasets are consistent. We therefore focus here on a subset of 32,467 OE62 molecules that contain the same elements as QM9 (H, C, O, N, and F). Furthermore, the original QM9 data was computed at the B3LYP/6-31G(2df,p) level, whereas the OE62 database is based on the Perdew-Burke-Ernzerhof (PBE) functional with Tkatchenko-Scheffler Van-der-Waals correction (PBE-vdW), tight integration grids and a "tier2" basis set of numerical atomic orbitals.[33–35] To increase the consistency between both datasets, the atomization energies for all QM9 molecules were correspondingly recomputed with the OE62 settings (using the original QM9 geometries). This new dataset is freely available from the authors.

Hyperparameter Optimization: The hyperparameters σ and λ (from Eqs. 2 and 3) were optimized through 4-fold cross validation (CV). Specifically, the parameters that minimize the average root mean square difference (RMSD) in CV were obtained using the Nelder-Mead minimization algorithm[36,37] as implemented in the scikit-learn package.[38] MBTR vectors were obtained via the *DScribe* package, including only one- and two-body terms.[39] Unlike σ and λ, the MBTR-specific hyperparameters were not optimized, and the default values for broadening and damping functions were used (see SI).

We note that using higher order terms and optimizing all hyperparameters would certainly lead to somewhat lower errors. However, the goal of this study is not to benchmark MBTR itself but to understand the role of size-extensivity on ML models with global representations. For this purpose, we found the above choices to be adequate.

4. Results and discussion

As discussed in the theory section, we will focus on three KRR models, namely the combinations MBTR/Gaussian, iMBTR/Gaussian and MBTR/linear. In line with previous ML studies on predicting AEs, we start by checking the predictive performance of the models within a dataset.[2,6,18] Here, we focus on a subset of QM9, containing all 3,993 molecules with up to seven heavy atoms. The average RMSD from 4-fold CV on this set is shown in Table 1.

The MBTR/Gaussian kernel performs best, followed by the iMBTR/Gaussian and MBTR/linear models. This shows the benefit of the non-linear Gaussian kernel, though the results of the linear kernel are also respectable, in line with what was reported by Huo and Rupp.[18] For consistency, all errors are reported with respect to total AEs, even for the iMBTR/Gaussian

Table 1. Averaged RMSD from 4-fold cross validation KRR models trained on the 3,993 QM9 molecules with HAC = 1–7.

MBTR Normalization	Kernel	Training Target	RMSD (kcal/mol)
iMBTR	Gaussian	AE/N	3.14
MBTR	linear	AE	4.09
MBTR	Gaussian	AE	2.30

model which is trained on AE/N. It is therefore not surprising that iMBTR/Gaussian performs somewhat more poorly than MBTR/Gaussian, given that it minimizes a different loss function. Still, one might naively conclude from this analysis that the conventional MBTR/Gaussian kernel is suitable for predicting AEs, in spite of its lacking extensivity.

This picture changes radically when the models are forced to extrapolate beyond the scope of their training sets, however. To this end, we consider a separate test set of 2000 QM9 molecules with nine heavy atoms. In addition to the standard HAC = 1–7 training set, we thereby also consider training sets containing only up to four, five, and six heavy atoms, respectively, to specifically test the extrapolation capabilities of the models. The results for all models are summarized in Figure 3.

Contrary to the previous result, the original MBTR/Gaussian method now shows the worst prediction performance among the three models, which is a direct manifestation of its lacking size-extensivity. Even the (extensive) MBTR/linear model shows significantly lower RMSD compared to MBTR/Gaussian. Finally, the iMBTR/Gaussian model combines proper extensivity with the non-linearity of the Gaussian kernel and performs best. Indeed, it even provides qualitatively useful predictions (with a relative error of ca. 1–2%) for the smallest training set, which consists of just 48 molecules with up to four heavy atoms.

An even more challenging test case is predicting the AEs of the OE62 set while still training on QM9. As mentioned above, the latter has a very narrow heavy atom distribution (peaking at 9) whereas the former has a wide distribution peaking around 20 (see Figure S1 in the SI). Furthermore, OE62 contains chemical structures that are absent from QM9, such as polycyclic aromatic compounds. As before, the models are trained on the 3,993 QM9 molecules with up to seven heavy atoms.

The correlations between predicted and reference AEs for all KRR models are shown in Figure 4. Here, the most notable feature is the abysmal performance of MBTR/Gaussian, with an RMSD of 4,327 kcal/mol. While the model actually displays reasonable accuracy up to AEs around ca. 2,500 kcal/mol (*i.e.* for molecules similar to the training set), it completely fails beyond this range. Indeed, as the kernel function vanishes for large molecules, the model predicts an AE of zero for all large molecules. This poor performance of MBTR/Gaussian vividly demonstrates its lack of size-extensivity.

In contrast, the iMBTR/Gaussian and MBTR/linear models both show good correlations with the reference across the full range of systems ($R^2 = 0.99$), with dramatically lower RMSDs of 184.4 and 138.2 kcal/mol, respectively. At first glance, this is still a large margin of error, compared to the results for QM9. It should however be noted that the error of a predicted AE should itself be size-extensive, so that larger errors are to be expected for larger systems.[21] Given that the AEs of the OE62 set range up to ca. 18,000 kcal/mol, an RMSD of ca. 100 kcal/mol is actually not that poor in relative terms. To quantify this, the RMSD can be normalized by the standard deviation of the Aes in the data set. This yields normalized RMSDs of 0.10 (iMBTR/Gaussian) and 0.08 (MBTR/linear), respectively (where 1.0 would be the performance of a random Gaussian model with appropriate mean and standard deviation).

Furthermore, this error is quite systematic, with the AEs of large systems being consistently underestimated. A linear fit of the correlation plots reveals that this is a bit more pronounced for iMBTR/Gaussian than for MBTR/linear (see Figure 4). Indeed,

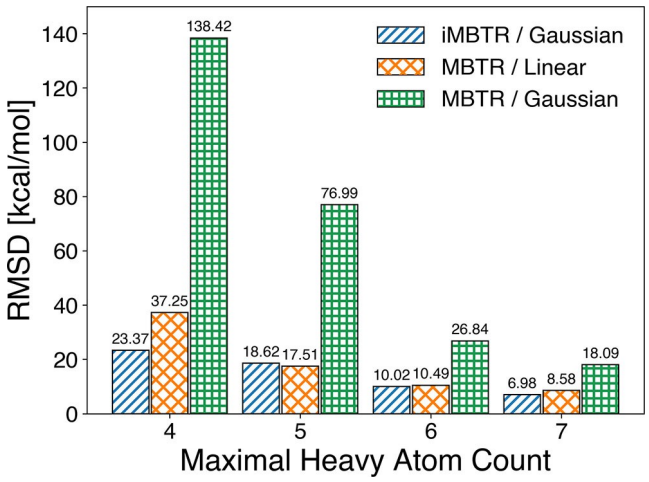


Figure 3. Accuracy of KRR models trained on small QM9 molecules (max. HAC = 4–7) when predicting larger molecules from QM9 (HAC = 9).

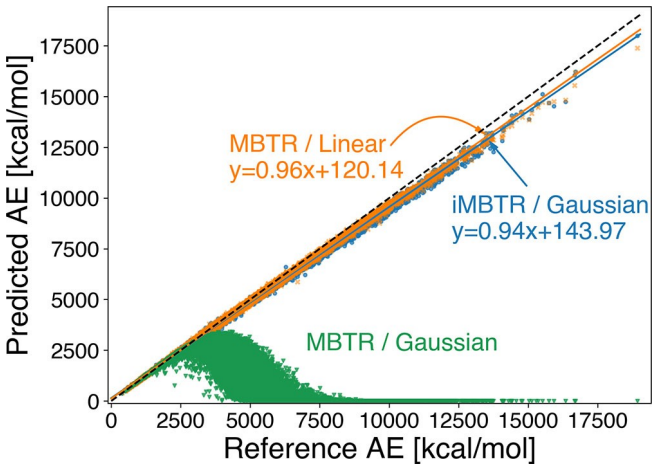


Figure 4. Correlation plots of predicted OE62 AEs for MBTR/Gaussian (▼ green), iMBTR/Gaussian (○ blue) and MBTR/linear (× orange). All models were trained on 3,993 QM9 molecules with HAC = 1–7. Prediction was performed on 32,467 OE62 molecules consisting of C, H, O, N and/or F. Linear regression lines and equations are shown for iMBTR/Gaussian (blue) and MBTR/linear (orange).

if the results of the linear regressions are subtracted from the predictions, the corresponding RMSDs are reduced to 63.85 kcal/mol (iMBTR/Gaussian) and 61.33 kcal/mol (MBTR/linear).

Of course, even in relative terms, the errors of these models are still larger than what would be expected purely based on the cross-validation RMSD of their training sets. This is because extensivity is not the only relevant size-effect. For example, long-range interactions like electrostatics and dispersion can play a significant role in stabilizing large molecules. Furthermore, electronic effects like quantum confinement may occur on the nanometer scale. These effects lead to a net stabilization of larger molecules, reflected in the systematic underestimation of the AEs mentioned above.

Consequently, AE/N is not converged for systems with seven heavy atoms, even in the fairly simple case of linear hydrocarbons (Figure 2). In Figure 5, the distribution of AE/N vs. N is shown for the full QM9 and OE62 sets. Interestingly, the basic features of this plot are remarkably similar to Figure 2. In particular, it can be seen that the mean AE/N is approximately constant for molecules with more than ca. 20 atoms. This regime corresponds to the largest molecules in QM9. The figure also provides an intuitive explanation of why the iMBTR/Gaussian method works. By choosing AE/N as the target quantity, the variability that the model must account for is decreased from ca. 18,000 kcal/mol to ca. 80 kcal/mol.

5. Conclusion

In this contribution, we have explored the size-extensivity of molecular ML models based on global representations such as the MBTR. While the conventional MBTR/Gaussian model is not ideal for either extensive or intensive properties, we showed that there are appropriate kernels for both cases, namely the MBTR/linear (extensive) and iMBTR/Gaussian (intensive). While current extensive ML models are typically built from local representations, our work shows that this is not strictly a requirement. We also showed how an intensive kernel can be used to predict an extensive property. To illustrate the significance of these results, a highly challenging ML task with large size differences between the molecules in the training and test sets was devised. We found that properly extensive models perform reasonably well in this setting, whereas the conventional MBTR/Gaussian approach fails outright.

Importantly, we stress that a non-extensive model can still be quite accurate if the size of the chemical space of interest is limited. However, in those areas of chemistry where ML is expected to have a large impact, this is not the case. In particular, for the study of large reaction networks (*e.g.* within systems chemistry or catalysis) a useful ML model must adequately describe the transition from small molecules to larger systems and even polymers (and *vice versa*). The present work represents an important stepping stone to this end.

Finally, it should be noted that the present study was purposefully designed to study the effects of size-extensivity in the limit of large size differences between training and test molecules. In practice, we expect that the systematic errors in the extensive models could be mitigated by including a limited number of larger molecules in the training set.

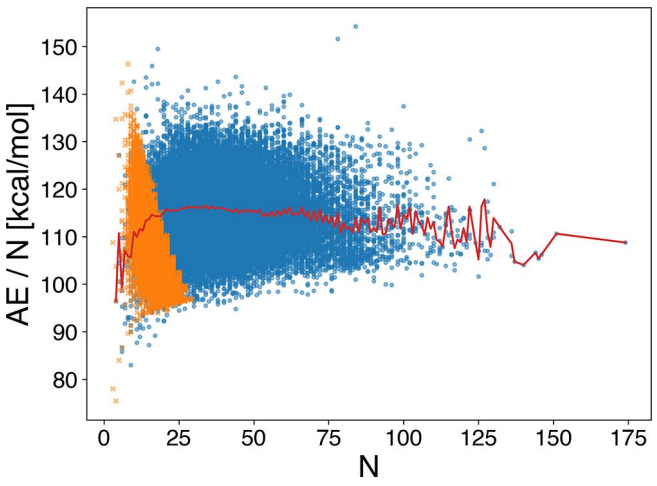


Figure 5. Plot of AE/N vs. N for molecules in QM9 (× orange) and OE62 (○ blue). The mean AE/N is shown as a red line.

Acknowledgements

This work was supported by the Korea Ministry of Environment (MOE) as "the Chemical Accident Prevention Technology Development Project" and the Global Frontier Program through the Global Frontier Hybrid Interface Materials (GFHIM) (2013-M3A6B1078882). We further acknowledge support by Deutsche Forschungsgemeinschaft (DFG) through TUM International Graduate School of Science and Engineering (IGSSE) (GSC 81). Computer time on resources of the Argonne Leadership Computing Facility was provided by the Innovative and Novel Computational Impact on Theory and Experiment (INCITE) program.

Conflict of Interest

The authors declare no conflict of interest.

Keywords: Machine learning · Kernel ridge regression · Many-body tensor representation · Size-extensivity · Atomization energy

[1] F. A. Faber, L. Hutchison, B. Huang, J. Gilmer, S. S. Schoenholz, G. E. Dahl, O. Vinyals, S. Kearnes, P. F. Riley, O. A. Von Lilienfeld, *J. Chem. Theory Comput.* **2017**, *13*, 5255–5264.
[2] K. Hansen, G. Montavon, F. Biegler, S. Fazli, M. Rupp, M. Scheffler, O. A. Von Lilienfeld, A. Tkatchenko, K.-R. Müller, *J. Chem. Theory Comput.* **2013**, *9*, 3404–3419.
[3] A. P. Bartók, S. De, C. Poelking, N. Bernstein, J. R. Kermode, G. Csányi, M. Ceriotti, *Sci. Adv.* **2017**, *3*, e1701816.
[4] M. Rupp, A. Tkatchenko, K.-R. Müller, O. A. Von Lilienfeld, *Phys. Rev. Lett.* **2012**, *108*, 58301.

[5] W. Pronobis, A. Tkatchenko, K.-R. Müller, *J. Chem. Theory Comput.* **2018**, *14*, 2991–3003.
[6] A. Stuke, M. Todorović, M. Rupp, C. Kunkel, K. Ghosh, L. Himanen, P. Rinke, *J. Chem. Phys.* **2019**, *150*, 204121.
[7] W. Pronobis, K. T. Schütt, A. Tkatchenko, K.-R. Müller, *Eur. Phys. J. B* **2018**, *91*, 178.
[8] R. Ramakrishnan, M. Hartmann, E. Tapavicza, O. A. Von Lilienfeld, *J. Chem. Phys.* **2015**, *143*, 84111.
[9] J. Kang, S. H. Noh, J. Hwang, H. Chun, H. Kim, B. Han, *Phys. Chem. Chem. Phys.* **2018**, *20*, 24539–24544.
[10] J. T. Margraf, K. Reuter, *ACS Omega* **2019**, *4*, 3370–3379.
[11] C. Kunkel, C. Schober, J. T. Margraf, K. Reuter, H. Oberhofer, *Chem. Mater.* **2019**, *31*, 969–978.
[12] C. Kunkel, C. Schober, H. Oberhofer, K. Reuter, *J. Mol. Model.* **2019**, *25*, 87.
[13] A. Bruix, J. T. Margraf, M. Andersen, K. Reuter, *Nat. Can.* **2019**, *2*, 659–670.
[14] J. A. Pople, M. Head-Gordon, D. J. Fox, K. Raghavachari, L. A. Curtiss, *J. Chem. Phys.* **1989**, *90*, 5622–5629.
[15] R. Peverati, D. G. Truhlar, *Philos. Trans. R. Soc. London* **2014**, *372*, 20120476.
[16] A. Karton, N. Sylvetsky, J. M. L. Martin, *J. Comput. Chem.* **2017**, *38*, 2063–2075.
[17] M. Rupp, *Int. J. Quantum Chem.* **2015**, *115*, 1058–1073.
[18] H. Huo, M. Rupp, *arXiv Prepr. arXiv1704.06439* **2017**.
[19] K. Hansen, F. Biegler, R. Ramakrishnan, W. Pronobis, O. A. Von Lilienfeld, K.-R. Müller, A. Tkatchenko, *J. Phys. Chem. Lett.* **2015**, *6*, 2326–2331.
[20] R. Ramakrishnan, P. O. Dral, M. Rupp, O. A. Von Lilienfeld, *Sci. Data* **2014**, *1*, 140022.
[21] J. T. Margraf, D. S. Ranasinghe, R. J. Bartlett, *Phys. Chem. Chem. Phys.* **2017**, *19*, 9798–9805.
[22] C. Schober, K. Reuter, H. Oberhofer, *J. Phys. Chem. Lett.* **2016**, *7*, 3973–3977.
[23] J. P. Perdew, J. Sun, A. J. Garza, G. E. Scuseria, *Z. Physiol. Chem.* **2016**, *230*, 737–742.
[24] R. Ramakrishnan, O. A. von Lilienfeld, *Chim. Int. J. Chem.* **2015**, *69*, 182–186.
[25] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, K.-R. Müller, *Sci. Adv.* **2017**, *3*, e1603015.
[26] J. Behler, M. Parrinello, *Phys. Rev. Lett.* **2007**, *98*, 146401.
[27] A. P. Bartók, M. C. Payne, R. Kondor, G. Csányi, *Phys. Rev. Lett.* **2010**, *104*, 136403.
[28] A. P. Bartók, R. Kondor, G. Csányi, *Phys. Rev. B* **2013**, *87*, 184115.
[29] S. Kondati Natarajan, J. Behler, *J. Phys. Chem. C* **2017**, *121*, 4368–4383.
[30] V. L. Deringer, G. Csányi, *Phys. Rev. B* **2017**, *95*, 094203.
[31] F. A. Faber, A. S. Christensen, B. Huang, O. A. von Lilienfeld, *J. Chem. Phys.* **2018**, *148*, 241717.
[32] K. Mills, K. Ryczko, I. Luchak, A. Domurad, C. Beeler, I. Tamblyn, *Chem. Sci.* **2019**, *10*, 4129–4140.
[33] V. Blum, R. Gehrke, F. Hanke, P. Havu, V. Havu, X. Ren, K. Reuter, M. Scheffl *Comput. Phys. Commun.* **2009**, *180*, 2175–2196.
[34] J. P. Perdew, K. Burke, M. Ernzerhof, *Phys. Rev. Lett.* **1996**, *77*, 3865–3868.
[35] A. Tkatchenko, M. Scheffler, *Phys. Rev. Lett.* **2009**, *102*, 073005.
[36] J. A. Nelder, R. Mead, *Comput. J.* **1965**, *7*, 308–313.
[37] M. H. Wright, *Pitman Res. Notes Math. Ser.* **1996**, 191–208.
[38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
[39] L. Himanen, M. O. J. Jäger, E. V. Morooka, F. Federici Canova, Y. S. Ranawat, D. Z. Gao, P. Rinke, A. S. Foster, *Comput. Phys. Commun.* **2019**, 106949.

Manuscript received: October 23, 2019
Version of record online: February 4, 2020

ChemSystemsChem

Supporting Information

Size-Extensive Molecular Machine Learning with Global Representations**

Hyunwook Jung[+], Sina Stocker[+], Christian Kunkel, Harald Oberhofer, Byungchan Han, Karsten Reuter, and Johannes T. Margraf*

Supporting Information:

Size-Extensive Molecular Machine Learning with Global Representations

Hyunwook Jung[†1,2], Sina Stocker[†1], Christian Kunkel[1], Harald Oberhofer[1], Byungchan Han[2], Karsten Reuter[1], and Johannes T. Margraf[1,*]

[1]Chair for Theoretical Chemistry and Catalysis Research Center, Technische Universität München, Lichtenbergstraße 4, D-85747 Garching, Germany

[2]Department of Chemical and Biomolecular Engineering, Yonsei University, Seoul 03722, Republic of Korea

[*]Corresponding author: Johannes T. Margraf, johannes.margraf@ch.tum.de

December 20, 2019

[†]These authors contributed equally to the manuscript

Table S1: MBTR-specific hyperparameters setting for Many-Body Tensor Representation (MBTR)

k-body	Geometry	Grid				Weighting		
	Function	Min	Max	n	σ	Function	Scale	Cutoff
k=1	Atomic number	0	10	100	0.1		-	
k=2	Inverse distance	0	1	100	0.1	Exponential	0.5	1e-3

Table S2: Optimized hyperparameters for three Kernel methods for different training set size

MBTR Normalization	Kernel	Training Target	HAC	RMSD (kcal/mol)	σ	$\log_2\lambda$
iMBTR	Gaussian	AE/N	7	3.14	1.884	-18.641
			6	3.87	2.679	-16.409
			5	6.41	3.634	-15.632
			4	11.63	8.674	-14.196
MBTR	Linear	AE	7	4.09	-	-1.093
			6	3.98	-	-2.915
			5	19.49	-	-0.956
			4	6.312	-	-57.365
MBTR	Gaussian	AE	7	2.304	217.959	-19.094
			6	3.323	357.352	-20.348
			5	16.313	392.877	-16.209
			4	10.827	393.283	-17.537

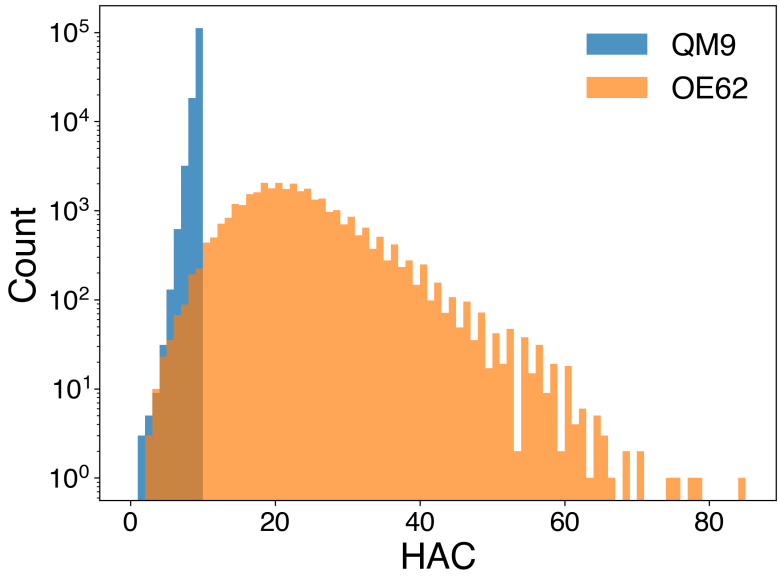


Figure S1: Histogram of heavy atom distribution in QM9 and OE62 dataset

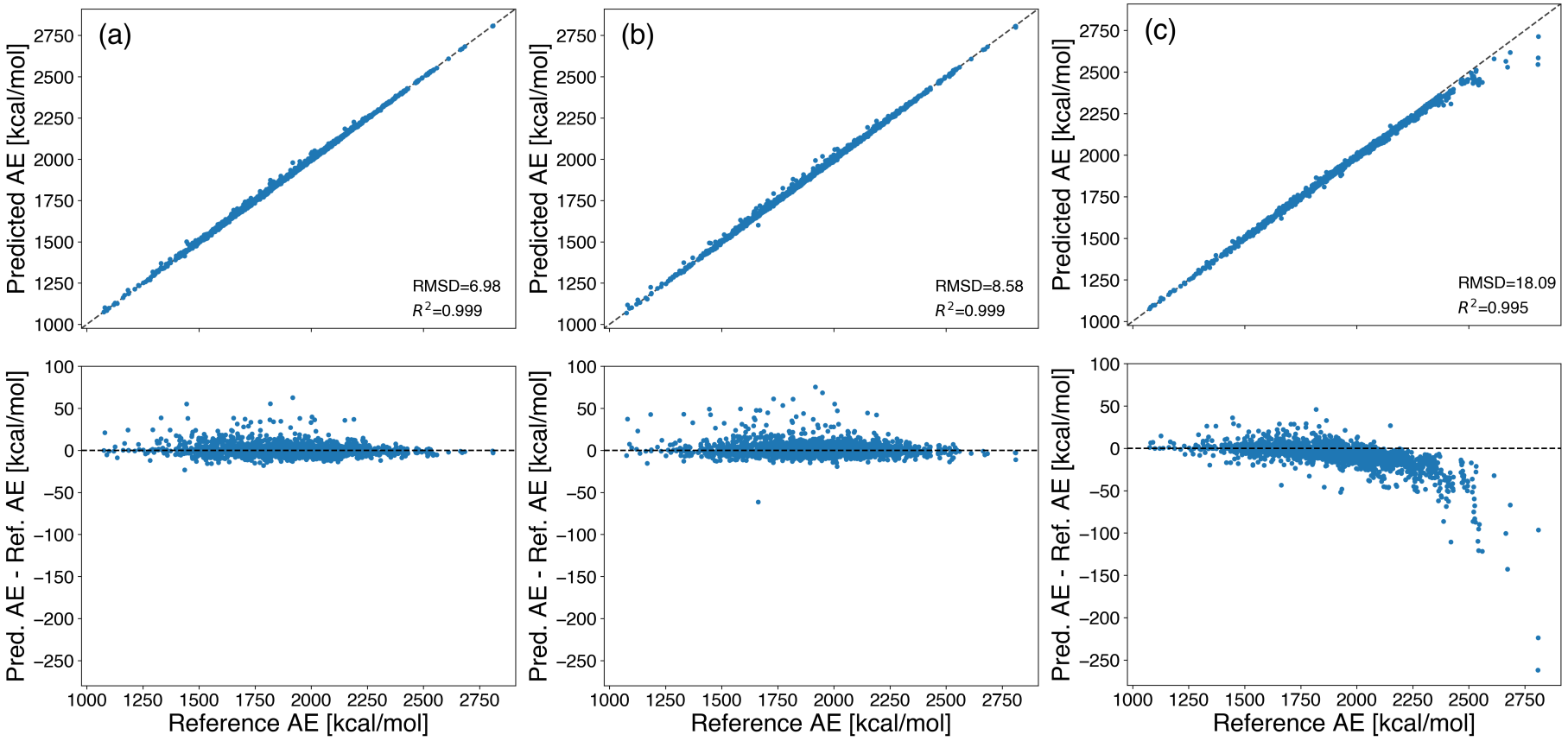


Figure S2: Correlation plot of linear correlation between predicted AE and reference DFT calculated AE. Trained on small molecules up to 7 heavy atoms and evaluated on 2000 randomly sampled molecules with 9 heavy atoms: (a) iMBTR/Gaussian, (b) MBTR/linear, (c) MBTR/Gaussian

Paper # 3

How Robust are Modern Graph Neural Network Potentials in Long and Hot Molecular Dynamics Simulations?

Sina Stocker*, Johannes Gasteiger*, Florian Becker, Stephan Günnemann and Johannes T. Margraf

Published on ChemRxiv. Cambridge: Cambridge Open Engage, (2022).
DOI: https://doi.org/10.26434/chemrxiv-2022-mc4gb‡
Submitted for publication to Machine Learning: Science and Technology.§

*These authors contributed equally
‡This content is a preprint.
§An updated version of the peer-reviewed and accepted paper is now publicly available on DOI: https://doi.org/10.1088/2632-2153/ac9955.

How Robust are Modern Graph Neural Network Potentials in Long and Hot Molecular Dynamics Simulations?

Sina Stocker,[1, 2, a)] Johannes Gasteiger,[2, a)] Florian Becker,[2] Stephan Günnemann,[2] and Johannes T. Margraf[1, b)]
[1)] *Fritz-Haber-Institute of the Max-Planck-Society, Germany*
[2)] *Technical University of Munich, Germany*

(Dated: 21 March 2022)

Graph neural networks (GNNs) have emerged as a powerful machine learning approach for the prediction of molecular properties. In particular, recently proposed advanced GNN models promise quantum chemical accuracy at a fraction of the computational cost. While the capabilities of such advanced GNNs have been extensively demonstrated on benchmark datasets, there have been few applications in real atomistic simulations. Here, we therefore put the robustness of GNN interatomic potentials to the test, using the recently proposed GemNet architecture as an example. Models are trained on the QM7-x database of organic molecules and used to perform extensive MD simulations. We find that low test set errors are not sufficient for obtaining stable dynamics and that severe pathologies sometimes only become apparent after hundreds of ps of dynamics. Nonetheless, highly stable and transferable GemNet potentials can be obtained with sufficiently large training sets.

Atomistic simulations are an invaluable tool for gaining mechanistic and structural insight into chemical systems, including solid materials[1], interfaces[2,3], liquids[4] or even complex biological systems like the SARS-CoV-2 virus[5]. They are also becoming increasingly important in the design of new materials and drugs[6,7]. In many ways, the prototypical atomistic simulation is a Molecular Dynamics (MD) trajectory, which propagates the atomic coordinates of a system in time, starting from some initial conditions. MD simulations are extremely common, both by themselves and as part of more elaborate sampling procedures like parallel tempering or metadynamics.

In principle, highly accurate MD trajectories can be obtained from electronic structure methods like density functional theory (DFT). Unfortunately, such *ab initio* MD (AIMD) simulations require the (approximate) solution of the electronic Schrödinger equation at every time step. This makes them very expensive from a computational perspective and ultimately limits the applicability of AIMD to a few hundreds of atoms and relatively short (*i.e* ps) timescales. For many scientific questions, simulations of much larger systems, longer timescales or (usually) both are required. To this end, empirical interatomic potentials are typically used. These provide an analytical expression for high-dimensional potential energy surfaces which can be evaluated in a small fraction of the time required for a DFT calculation. This gain in efficiency invariably comes at the expense of a decrease in accuracy and/or transferability, however.

To bridge this gap between computational cost and accuracy, machine learned interatomic potentials have recently gained popularity in computational chemistry[8–11] and materials science[12–14]. In particular, a range of neural network[15–17] and kernel based potentials[18,19] have been developed and applied to a wide variety of chemical systems. While somewhat more expensive than classical force fields, these potentials are able to predict energies and forces with DFT accuracy and have thus become an important part of the toolbox of computational chemistry.

One of the most recent additions to this family of methods are potentials based on graph neural networks (GNNs), such as SchNet, DimeNet, GemNet and NequIP.[20–31] Here, much progress towards ever more accurate and expressive potentials has been made, e.g. by using equivariant formulations or embedding atom pairs and triplets. While such efforts naturally focus on established benchmark databases like QM9[32,33], MD17[34] or OC20[35], comparatively little research has been conducted to show the applicability of such advanced GNN potentials in real atomistic simulations. A notable exception to this is a recent paper of Batzner and coworkers[22], which demonstrated that potentials based on the equivariant NequIP architecture could be used in stable and accurate MD simulations, when trained on AIMD data for the respective system.

In this contribution, we aim to provide an in-depth exploration of the robustness of state-of-the-art GNN potentials based on the GemNet architecture[21] in MD simulations. To this end, we ran a total of 245 ns of dynamics (around 500 million timesteps) across a wide range of temperatures and organic molecules. By checking samples from these large ensembles with DFT reference calculations, the extrapolative capabilities of these potentials in configuration and chemical space was tested simultaneously. Furthermore, the impact of training set size on the robustness of the potentials was explored.

GNNs treat chemical systems as graphs, with nodes representing atoms and edges representing interactions between atom pairs. While traditional chemical graph representations usually equate edges with covalent bonds, GNNs assume edges between all atoms within a given cutoff. All potentials discussed in the following are based on the geometric message passing neural network (GemNet)[21], which shows excellent performances on established benchmark data sets like MD17 and OC20 as well as QM7-x (see Fig. 1). GemNet embeds both the atoms and the interatomic edges via high-dimensional

[a)] The first two authors contributed equally
[b)] Electronic mail: margarf@fhi.mpg.de

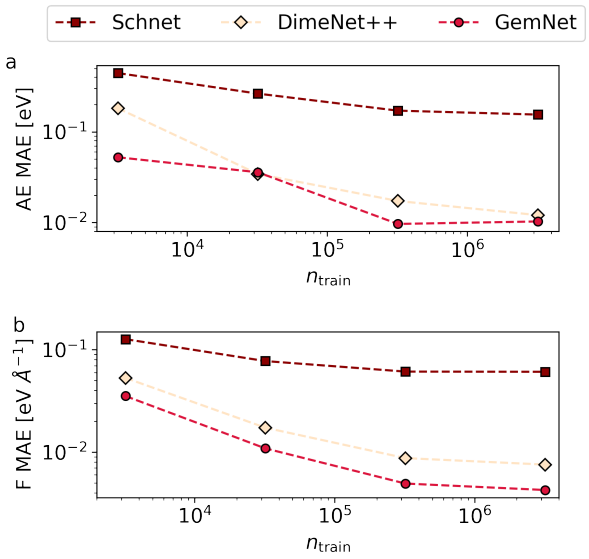


FIG. 1: **Learning curves.** (a) Mean absolute errors (MAEs) of atomization energies (AE) and (b) forces (F) against the number of training configurations. MAEs are calculated for a test set consisting of 10,100 random configurations from the QM7-x database.

vectors. Both kinds of embeddings are then updated in multiple layers using learnable weight matrices and by passing messages between the edges and atoms within a given cutoff distance. GemNet leverages the full geometric information for this: The interatomic distances, the angles between neighboring edges, and the dihedral angles defined via triplets of edges. From the learned embeddings, energy contributions for each atom and layer are obtained, which are subsequently summed up to calculate the total energy of the system. The whole model is continuously differentiable, which allows calculating the forces via $\mathbf{F}_i = -\frac{\partial}{\partial \mathbf{x}_i} E$. As for all GNNs, the use of a finite cutoff and per-atom energy contributions makes the predictions size-extensive and the computational cost scale linearly with the number of atoms.

Herein, we trained several GemNet potentials on different subsets of the recently published QM7-x database.[36] This dataset consists of around 4.2 million configurations sampled from small organic molecules consisting of up to seven non-hydrogen atoms (*i.e.* C, O, N, S, Cl), with 4-23 atoms in total. Importantly, QM7-x covers both equilibrium and non-equilibrium structures. Starting from 6,950 structural formulas, it contains around 41,500 equilibrium structures (including stereoisomers and conformers) and 100 additional non-equilibrium structures for each equilibrium geometry. The latter were generated by applying linear combinations of normal mode displacements to each configurations, thus approximately mimicking molecular dynamics within the harmonic approximation. For each configuration, total energies and forces at the hybrid DFT (PBE0)[37] level with a many-body dispersion correction (MBD)[38] are provided, computed with tightly converged numerical atom-centered basis sets and integration grids[39,40] (see Ref. [36] for full details).

GemNet potentials were trained on atomization energies (AE) and forces (F) simultaneously. Since forces are ultimately the driver of MD simulations and contain more fine-grained information than energies, forces were weighted more strongly in our fits, so that the AE only contributes 0.1% to the loss function (see SI for details). This essentially follows the philosophy of gradient domain machine learning,[34,41] which exclusively uses forces. However, we do include a small AE contribution to the loss, as energy differences across chemical space cannot be learned effectively from forces alone.[42] For training, the QM7-x dataset was randomly split into a test set of 10,100 configurations, training sets of 3.2k, 32k, 320k and 3.2Mio configurations and corresponding validation sets of 800, 8k, 80k and 800k configurations (the latter being used for hyperparameter selection, see SI). In the interest of simplification, we will denote models trained on small (3.2k and 32k) and large (320k and 3.2Mio) training sets as 'sparse' and 'exhaustive' models respectively.

In Fig. 1, the corresponding learning curves for AE and F are shown. The force curve shows a roughly linear decrease on the log-log scale between 3.2k and 320k training configurations but levels off between 320k and 3.2Mio configurations. This indicates that the more exhaustive models approach the intrinsic accuracy that is possible given the precision of the data and limitations of the models themselves (e.g. due to the cutoffs employed). Due to the lower weighting of energies in the loss the AE curve is somewhat more noisy but follows the same trend.

To put this performance into perspective, the most exhaustive GemNet model yields a force MAE of 0.0043 eV Å^{-1}, which can be compared with an MAE of 0.015 eV Å^{-1} for the recently developed SpookyNet[23] architecture (in this case trained on 4.2Mio molecules). In addition, GemNet outperforms SchNet[28] and DimNet++[25] on QM7-x for nearly all points of the learning curve (with the only exception being the AE error of the 32k model). Importantly, the energy errors are very low (0.01 eV = 0.23 kcal mol^{-1}) despite the low weighting of AEs in the loss. It is furthermore notable that even the model trained on 3.2k configurations displays quite good performance with MAEs of around 0.035 eV Å^{-1} and 0.05 eV ($\approx$ 1 kcal mol^{-1}).

To explore the robustness of the GemNet potentials within the scope of their training set, constant temperature MD simulations were performed for 20 representative molecules from QM7-x (see FIG. 2a). Here, care was taken to include all atom types in the dataset. For each molecule, 1 ns trajectories were generated with a 0.5 fs timestep at three different temperatures (300 K, 600 K and 1200 K), using all models presented in the learning curve (see SI for details on the MD simulations). The rationale for using these temperatures is that they lead to increasingly extensive exploration of phase space. Indeed, it is not uncommon to use high temperature dynamics for this purpose, *e.g.* in replica exchange MD.[43] From each trajectory, 72 configuration were uniformly sampled and the corresponding energies and forces computed with identical DFT settings to the ones used for the QM7-x set.

Figures 2b and 2c show the AE and F MAEs for these

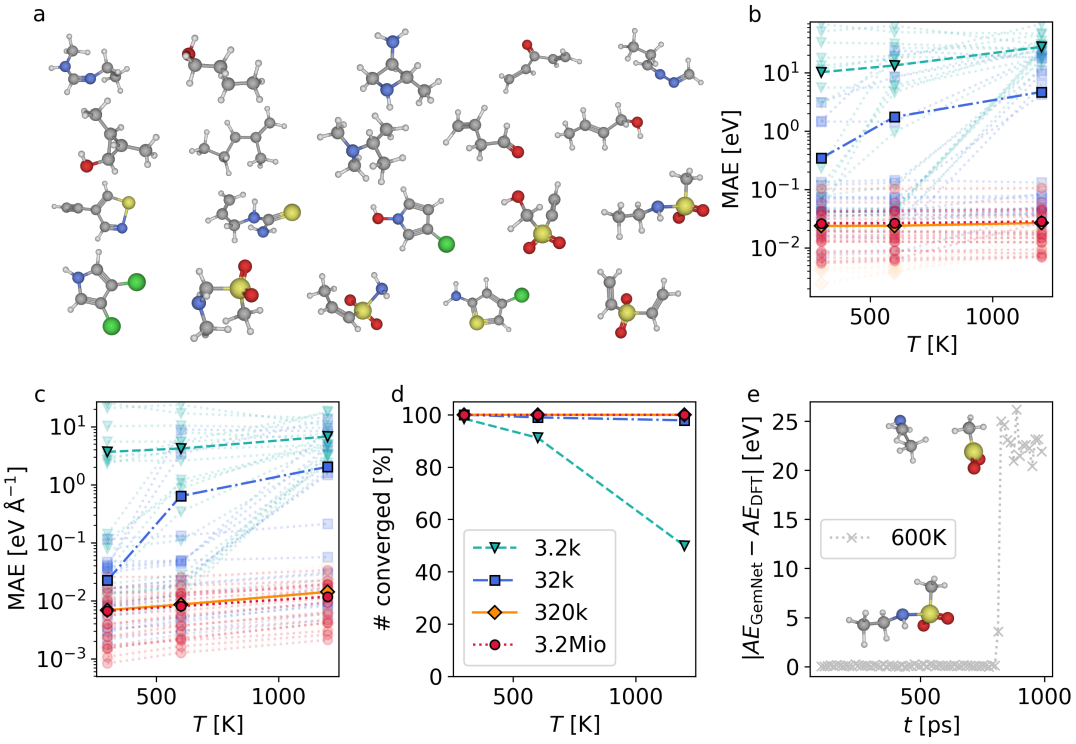


FIG. 2: **Robustness of GemNet potentials in molecular dynamics (MD).** (a) Representative molecules from QM7-x used in the MD tests. (b), (c) Mean absolute errors (MAEs) on atomization energy (AE) and force (F) predictions as a function of training set size and temperature. Opaque lines and symbols represent the average MAE over all molecules in (a). Transparent lines and symbols represent the MAE of one specific molecule. (d) Percentage of converged DFT calculations for configurations generated with different potentials and temperatures. (e) AE error as a function of simulation time for a 1 ns MD trajectory at 600 K, using a potential trained with 3.2k configurations. This sub-panel shows a drastic deterioration in energy predictions after around 700 ps, when the molecule dissociates into fragments that cannot be accurately described by the model.

samples as a function of temperature and training set size. Here, opaque symbols and lines represent MAEs averaged over 20 different trajectories corresponding to a given model and temperature. Transparent symbols and lines illustrate the MAEs for each trajectory individually, to provide some insight into the spread of MAEs for different molecules (see SI for additional illustrations of the respective error distributions). Overall, we find quite consistent trends for both AE and F predictions. Whereas the exhaustive models (320k and 3.2Mio) only display a very slight increase of the MAEs with temperature, the errors of the sparse models (3.2k and 32k) increase dramatically. This is expected, as higher temperature MD simulations more extensively explore the phase space and consequently move away from the training configurations.

Notably, the 3.2k model already displays a very large AE error of more than 10 eV at 300 K. The MD error is thus orders of magnitude larger than the test set error, even though these configurations should arguably fall within the scope of the training set. This mainly stems from the fact that the trajectories for certain molecules lead to completely unphysical configurations, for which the potential then displays extremely large errors. Such unphysical configurations also commonly lead to convergence issues in the reference DFT calculations. To quantify this, the percentage of converged DFT calculations for configurations obtained with a given potential and simulation temperature is shown in Fig. 2d. We find that all DFT calculations converge for the 320k and 3.2Mio potentials, while the sparse models generate increasingly unphysical configurations with rising temperature. This is particularly evident for the 3.2k model at 1200 K, where only about half of the DFT calculations converge.

The marked discrepancy between the test set and MD performance of the 3.2k model underscores the limitations of using test configurations that are not generated by the potential itself. For a ML model to be useful in atomistic simulations, it is not sufficient to show that it provides accurate fits for physically reasonable configurations. It is equally important that the model avoids unphysical configurations in its own simulations. Note that testing this requires sufficiently long trajectories. This is illustrated for a representative example in Fig. 2e. Here, the error of the 3.2k model is actually quite low for the first 700 ps of the simulation after which it rises sharply to more than 20 eV due to an unphysical bond dissociation event. This behaviour can be understood as a kind

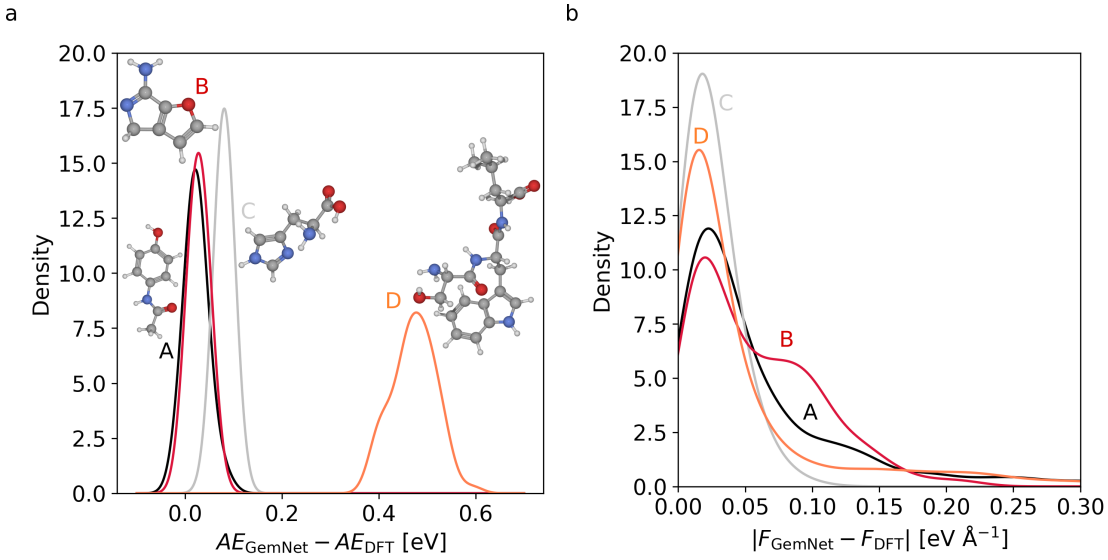


FIG. 3: **Out-of-sample validation of the GemNet potential trained on 3.2Mio configurations.** (a) Kernel density estimates of atomization energy (AE) error distributions for four out-of-sample molecules (A=Paracetamol: red, B=4H-Furo[2,3-c]pyrrol-6-amin: black, C=Histidin: grey and D=Ser-Trp-Leu-tripeptide: orange), obtained at 1200 K using the 3.2Mio potential. (b) Kernel density estimates of the corresponding force (F) error distributions.

of 'hole' in the potential energy surface of the model. This hole can be rather small, but once the simulation reaches such a configuration the trajectory is completely unreasonable. The 'robustness' of a ML potential can thus be understood as a measure of how frequent and how large such holes in the potential energy surface are. Ultimately this can only be quantified by performing MD simulations with the corresponding potential.

It should be stressed that this notion of robustness is not necessarily correlated with the test MAE, despite the fact that the robust GemNet models also display much lower MAEs. Indeed, the robustness of traditional bio-organic forcefileds with fixed topologies is very high. However, in this case robustness is gained at the expense of model flexibility. The challenge for ML potentials is that they must be robust without sacrificing flexibility. Our tests show that this is not trivial. On a more positive note, we do find that GemNet potentials with sufficiently large training sets are very robust across the phase space of the QM7-x dataset and beyond.

Another way to illustrate this is to consider the performance of the 3.2k model for a trajectory generated with the 3.2Mio potential in comparison with its own trajectory. Specifically this means that we generate two independent trajectories with the 3.2k and 3.2Mio model and evaluate MAEs of the 3.2k model for configurations drawn from each trajectory. Taking the molecule in Fig. 2e at 1200 K, the F MAE of the 3.2k potential is 6.8 eV Å^{-1} for the 3.2k trajectory but only 0.16 eV Å^{-1} when it is evaluated on the 3.2Mio trajectory. Again, the sparse model performs quite well for the physically reasonable configurations generated with the 3.2Mio model. The problem only becomes apparent when testing the sparse model on its own trajectory.

Having established the robustness of the exhaustive models within the scope of QM7-x, we now turn to the simultaneous exploration in chemical and configuration space. To this end, we consider four molecules consisting of 9-29 heavy atoms (*i.e.* which are significantly larger than the training molecules). Again, 1 ns MD trajectories were generated with the 3.2Mio potential at 1200 K. Figure 3 shows the corresponding AE and F error distributions. Strikingly, the AE errors are systematically more positive than the DFT reference energies, most prominently for the large Ser-Trp-Leu tripeptide. Here, the mean AE is shifted by 0.47 eV with respect to the reference, which is substantial when compared to an MAE of 0.0284 eV at 1200 K in Fig. 2b.

This shift can be explained by the absence of attractive long-range interactions (*e.g.* dispersion or electrostatics) in the GemNet potential. While message-passing neural networks can in principle include information from beyond their cutoff distance, the QM7-x database exclusively consists of small molecules so that long-range interactions simply cannot be learned from it. Methods to include long-range interactions are proposed in literature[23,27,44,45] and could also be applied to the GemNet architecture. Nonetheless, GemNet and DFT energies are highly correlated ($R^2 = 0.998$, see SI) and the standard deviation of the AE error distribution is only 0.045 eV so that the MD trajectory for this system should still be considered to be of high quality. While the long-range interactions are thus considerable in magnitude, they do not fluctuate very strongly.[46] This is also the case for the other molecules, which display very narrow AE error distributions. Similarly, force component errors are consistently small, with MAEs between even 0.012 eV Å^{-1} and 0.036 eV Å^{-1}.

In conclusion, we have explored the robustness of GNN potentials based on the recent GemNet architecture in

MD simulations. We find that sufficiently large training sets are key to obtaining robust GNN potentials and that a low test set error does not guarantee that stable trajectories can be generated. Interestingly, in some cases severe instabilities were only discovered after hundreds of ps of dynamics. The test set error should thus not be taken at face value as a measure for the error one can expect in 'real' applications. Demonstrating 'chemical accuracy' on a test set is by itself not enough.

With large enough training sets, the GemNet potentials used herein do display impressive performance, however. This is demonstrated by applications in high-temperature MD simulations of systems that are significantly larger than the training molecules. In this extrapolative regime, errors are mostly systematic and explainable and no instabilities were observed. Interestingly, no significant improvements in terms of accuracy or robustness were observed when training on 3.2Mio instead of 320k samples, indicating that all relevant information about the underlying potential energy surface can be learned from less than 10% of the dataset. This is significant because robust ML potentials are often associated with iterative training procedures. Due to their size and complexity (the models used herein fit 2.2 million parameters), GNN models are *a priori* not ideal for such settings. Indeed, training times of several GPU weeks are not unusual, which is clearly impractical in an iterative workflow. Well curated databases like QM7-x and powerful model architectures like GemNet circumvent this issue.

As a final point, we note that the potentials discussed herein (as well as the underlying code) are freely available at `https://www.daml.in.tum.de/gemnet`. We recommend the 3.2Mio GemNet potential as a general-purpose force field for exploring the conformational space of small to medium organic molecules. Indeed, the accuracy and the robustness of the 320k and 3.2Mio models is high enough that they can be considered as a cost effective replacement of DFT calculations for this application. It remains to be seen whether equally accurate and robust models can be obtained for larger chemical spaces, broader sections of the periodic table and chemical reactions.

ACKNOWLEDGMENTS

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) through TUM International Graduate School of Science and Engineering (IGSSE) (GSC 81)

[1] V. L. Deringer, N. Bernstein, G. Csányi, C. Ben Mahmoud, M. Ceriotti, M. Wilson, D. A. Drabold, and S. R. Elliott, "Origins of structural and electronic transitions in disordered silicon," *Nature*, vol. 589, no. 7840, pp. 59–64, 2021.

[2] S. Stegmaier, R. Schierholz, I. Povstugar, J. Barthel, S. P. Rittmeyer, S. Yu, S. Wengert, S. Rostami, H. Kungl, K. Reuter, R.-A. Eichel, and C. Scheurer, "Nano-scale complexions facilitate li dendrite-free operation in latp solid-state electrolyte," *Adv. Energy Mater.*, vol. 11, no. 26, p. 2100707, 2021.

[3] J. Timmermann, F. Kraushofer, N. Resch, P. Li, Y. Wang, Z. Mao, M. Riva, Y. Lee, C. Staacke, M. Schmid, C. Scheurer, G. S. Parkinson, U. Diebold, and K. Reuter, "iro$_2$ surface complexions identified through machine learning and surface investigations," *Phys. Rev. Lett.*, vol. 125, p. 206101, 2020.

[4] B. Cheng, G. Mazzola, C. J. Pickard, and M. Ceriotti, "Evidence for supercritical behaviour of high-pressure liquid hydrogen," *Nature*, vol. 585, no. 7824, pp. 217–220, 2020.

[5] M. I. Zimmerman, J. R. Porter, M. D. Ward, S. Singh, N. Vithani, A. Meller, U. L. Mallimadugula, C. E. Kuhn, J. H. Borowsky, R. P. Wiewiora, M. F. D. Hurley, A. M. Harbison, C. A. Fogarty, J. E. Coffland, E. Fadda, V. A. Voelz, J. D. Chodera, and G. R. Bowman, "Sars-cov-2 simulations go exascale to predict dramatic spike opening and cryptic pockets across the proteome," *Nat. Chem.*, vol. 13, no. 7, pp. 651–659, 2021.

[6] M. Zhong, K. Tran, Y. Min, C. Wang, Z. Wang, C.-T. Dinh, P. De Luna, Z. Yu, A. S. Rasouli, P. Brodersen, S. Sun, O. Voznyy, C.-S. Tan, M. Askerka, F. Che, M. Liu, A. Seifitokaldani, Y. Pang, S.-C. Lo, A. Ip, Z. Ulissi, and E. H. Sargent, "Accelerated discovery of co2 electrocatalysts using active machine learning," *Nature*, vol. 581, no. 7807, pp. 178–183, 2020.

[7] N. Artrith, A. Urban, and G. Ceder, "Efficient and accurate machine-learning interpolation of atomic energies in compositions with many species," *Phys. Rev. B*, vol. 96, no. 1, pp. 014112–014112, 2017.

[8] O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schütt, A. Tkatchenko, and K.-R. Müller, "Machine learning force fields," *Chem. Rev.*, vol. 121, no. 16, pp. 10142–10186, 2021.

[9] J. A. Keith, V. Vassilev-Galindo, B. Cheng, S. Chmiela, M. Gastegger, K.-R. Müller, and A. Tkatchenko, "Combining machine learning and computational chemistry for predictive insights into chemical systems," *Chem. Rev.*, vol. 121, no. 16, pp. 9816–9872, 2021.

[10] J. Xu, X.-M. Cao, and P. Hu, "Accelerating metadynamics-based free-energy calculations with adaptive machine learning potentials," *J. Chem. Theory Comput.*, vol. 17, no. 7, pp. 4465–4476, 2021. PMID: 34100605.

[11] S. Stocker, G. Csányi, K. Reuter, and J. T. Margraf, "Machine learning in chemical reaction space," *Nat. Commun.*, vol. 11, no. 1, p. 5505, 2020.

[12] J. Kim, D. Kang, S. Kim, and H. W. Jang, "Catalyze materials science with machine learning," *ACS Mater. Lett.*, vol. 3, no. 8, pp. 1151–1171, 2021.

[13] V. L. Deringer, A. P. Bartók, N. Bernstein, D. M. Wilkins, M. Ceriotti, and G. Csányi, "Gaussian process regression for materials and molecules," *Chem. Rev.*, vol. 121, no. 16, pp. 10073–10141, 2021.

[14] J. Behler and G. Csányi, "Machine learning potentials for extended systems: a perspective," *Eur. Phys. J. B*, vol. 94, no. 7, p. 142, 2021.

[15] J. Behler and M. Parrinello, "Generalized neural-network representation of high-dimensional potential-energy surfaces," *Phys. Rev. Lett.*, vol. 98, p. 146401, 2007.

[16] J. S. Smith, O. Isayev, and A. E. Roitberg, "Ani-1: an extensible neural network potential with dft accuracy at force field computational cost," *Chem. Sci.*, vol. 8, pp. 3192–3203, 2017.

[17] J. S. Smith, B. T. Nebgen, R. Zubatyuk, N. Lubbers, C. Devereux, K. Barros, S. Tretiak, O. Isayev, and A. E. Roitberg, "Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning," *Nat. Commun.*, vol. 10, no. 1, pp. 2903–2903, 2019.

[18] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, "Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons," *Phys. Rev. Lett.*, vol. 104, p. 136403, 2010.

[19] A. P. Bartók, R. Kondor, and G. Csányi, "On representing chemical environments," *Phys. Rev. B*, vol. 87, p. 184115, 2013.

[20] K. Schütt, O. Unke, and M. Gastegger, "Equivariant message passing for the prediction of tensorial properties and molecular spectra," in *ICML*, 2021.

[21] J. Gasteiger, F. Becker, and S. Günnemann, "Gemnet: Universal directional graph neural networks for molecules," in *NeurIPS*, 2021.

[22] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, "E(3)-

equivariant graph neural networks for data-efficient and accurate interatomic potentials," 2021.
[23] O. T. Unke, S. Chmiela, M. Gastegger, K. T. Schütt, H. E. Sauceda, and K.-R. Müller, "Spookynet: Learning force fields with electronic degrees of freedom and nonlocal effects," 2021.
[24] J. Gasteiger, J. Groß, and S. Günnemann, "Directional message passing for molecular graphs," in *ICLR*, 2020.
[25] J. Gasteiger, S. Giri, J. T. Margraf, and S. Günnemann, "Fast and uncertainty-aware directional message passing for non-equilibrium molecules," in *ML for Molecules workshop, NeurIPS*, 2020.
[26] C. W. Park, M. Kornbluth, J. Vandermause, C. Wolverton, B. Kozinsky, and J. P. Mailoa, "Accurate and scalable multi-element graph neural network force field and molecular dynamics with direct force architecture," 2020.
[27] O. T. Unke and M. Meuwly, "Physnet: A neural network for predicting energies, forces, dipole moments, and partial charges," *J. Chem. Theory Comput.*, vol. 15, no. 6, pp. 3678–3693, 2019.
[28] K. T. Schütt, P.-J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko, and K.-R. Müller, "Schnet: A continuous-filter convolutional neural network for modeling quantum interactions," in *NeurIPS*, 2017.
[29] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, "Quantum-chemical insights from deep tensor neural networks," *Nat. Commun.*, vol. 8, no. 1, pp. 13890–13890, 2017.
[30] K. T. Schütt, P. Kessel, M. Gastegger, K. A. Nicoli, A. Tkatchenko, and K.-R. Müller, "SchNetPack: A Deep Learning Toolbox For Atomistic Systems," *J. Chem. Theory Comput.*, vol. 15, no. 1, pp. 448–455, 2019.
[31] R. Zubatyuk, J. S. Smith, J. Leszczynski, and O. Isayev, "Accurate and transferable multitask prediction of chemical properties with an atoms-in-molecules neural network," *Sci. Adv.*, vol. 5, p. eaav6490, 2019.
[32] L. Ruddigkeit, R. van Deursen, L. C. Blum, and J.-L. Reymond, "Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17," *J. Chem. Inf. Model.*, vol. 52, no. 11, pp. 2864–2875, 2012. PMID: 23088335.
[33] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld, "Quantum chemistry structures and properties of 134 kilo molecules," *Sci. Data*, vol. 1, 2014.
[34] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller, "Machine learning of accurate energy-conserving molecular force fields," *Sci. Adv.*, vol. 3, no. 5, p. e1603015, 2017.
[35] L. Chanussot, A. Das, S. Goyal, T. Lavril, M. Shuaibi, M. Riviere, K. Tran, J. Heras-Domingo, C. Ho, W. Hu, A. Palizhati, A. Sriram, B. Wood, J. Yoon, D. Parikh, C. L. Zitnick, and Z. Ulissi, "Correction to the open catalyst 2020 (oc20) dataset and community challenges," *ACS Catal.*, vol. 11, no. 21, pp. 13062–13065, 2021.
[36] J. Hoja, L. Medrano Sandonas, B. G. Ernst, A. Vazquez-Mayagoitia, R. A. DiStasio Jr., and A. Tkatchenko, "Qm7-x, a comprehensive dataset of quantum-mechanical properties spanning the chemical space of small organic molecules," *Sci. Data*, vol. 8, no. 1, p. 43, 2021.
[37] J. P. Perdew, M. Ernzerhof, and K. Burke, "Rationale for mixing exact exchange with density functional approximations," *J. Chem. Phys.*, vol. 105, no. 22, pp. 9982–9985, 1996.
[38] A. Tkatchenko, R. A. DiStasio, R. Car, and M. Scheffler, "Accurate and efficient method for many-body van der Waals interactions," *Phys. Rev. Lett.*, vol. 108, no. 23, pp. 1–5, 2012.
[39] V. Blum, R. Gehrke, F. Hanke, P. Havu, V. Havu, X. Ren, K. Reuter, and M. Scheffler, "Ab initio molecular simulations with numeric atom-centered orbitals," *Comput. Phys. Commun.*, vol. 180, no. 11, pp. 2175–2196, 2009.
[40] X. Ren, P. Rinke, V. Blum, J. Wieferink, A. Tkatchenko, A. Sanfilippo, K. Reuter, and M. Scheffler, "Resolution-of-identity approach to Hartree–Fock, hybrid density functionals, RPA, MP2 and GW with numeric atom-centered orbital basis functions," *New J. Phys.*, vol. 14, no. 5, pp. 053020–053020, 2012.
[41] S. Chmiela, H. E. Sauceda, K.-R. Müller, and A. Tkatchenko, "Towards exact molecular dynamics simulations with machine-learned force fields," *Nat. Commun.*, vol. 9, no. 1, pp. 3887–3887, 2018.
[42] A. S. Christensen and O. A. von Lilienfeld, "On the role of gradients for machine learning of molecular energies and forces," *Mach. Learn. Sci. Technol.*, vol. 1, p. 045018, 2020.
[43] R. Petraglia, A. Nicolaï, M. D. Wodrich, M. Ceriotti, and C. Corminboeuf, "Beyond static structures: Putting forth REMD as a tool to solve problems in computational organic chemistry," *J. Comput. Chem.*, vol. 37, no. 1, pp. 83–92, 2016.
[44] T. W. Ko, J. A. Finkler, S. Goedecker, and J. Behler, "A Fourth-Generation High-Dimensional Neural Network Potential with Accurate Electrostatics Including Non-local Charge Transfer," *Nat. Commun.*, vol. 12, p. 398, 2021.
[45] C. Staacke, S. Wengert, C. Kunkel, G. Csányi, Reuter, K, and J. Margraf, "Kernel Charge Equilibration: Efficient and Accurate Prediction of Molecular Dipole Moments with a Machine-Learning Enhanced Electron Density Model," *Mach. Learn. Sci. Technol.*, vol. 3, p. 015032, 2022.
[46] C. Staacke, H. Heenen, C. Scheurer, G. Csányi, K. Reuter, and J. Margraf, "On the role of long-range electrostatics in machine-learned interatomic potentials for complex battery materials," *ACS Appl. Energy Mater.*, vol. 4, no. 11, pp. 12562–12569, 2021.

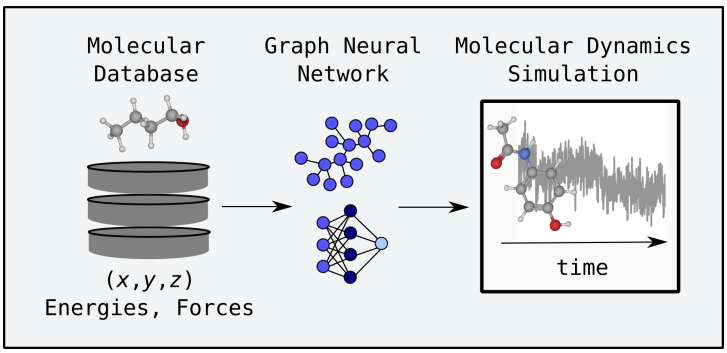


Table of Content Graphic

Supporting Information:

How Robust are Modern Graph Neural Network Potentials in Long and Hot Molecular Dynamics Simulations?

Sina Stocker,[1,2,a)] Johannes Gasteiger,[2,a)] Florian Becker,[2] Stephan Günnemann,[2] and Johannes T. Margraf[1,b)]

[1)] *Fritz Haber Institute of the Max-Plank society, Germany*

[2)] *Technical University of Munich, Germany*

[a)] The first two authors contributed equally

[b)] Electronic mail: margarf@fhi.mpg.de

I. COMPUTATIONAL METHODS

A. DFT reference calculations

We use the FHI-aims[1] (version 210713) code with the same computational settings as listed in ref[2] to perform reference calculations. DFT energies and forces have been validated for the 20 molecules illustrated in Fig. 1 of the main text, to ensure that we can reproduce energies and forces of the reference calculations.

B. GemNet

GNNs for molecules represent them as graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the atoms define the node set $\mathcal{V}$ and the interactions the edge set $\mathcal{E}$. GemNet creates edges for all atom pairs within a given cutoff of 5 Å. While regular message passing neural networks (MPNNs) only embed each atom a as $\boldsymbol{h}_a \in \mathbb{R}^H$,[3] GemNet additionally embeds the directed edges between atoms as $\boldsymbol{m}_{(ba)} \in \mathbb{R}^{H_\mathrm{m}}$. Each directed edge is associated with a direction in 3D space, pointing from atom b to atom a. These directions allow us to define angles from pairs of edges and dihedral angles from triplets of edges. Defining a dihedral angle requires four atoms in total: Two atoms are interacting (a and b) and two atoms define the messages (c and d). Message passing in GemNet is thus based on quadruplets of atoms.

GemNet transforms the geometric information using a set of radial, circular, and spherical basis functions (RBFs, CBFs, SBFs), which facilitates model accuracy. The distance between atoms b and a is transformed into $\boldsymbol{e}^{(ba)}_\mathrm{RBF}$, the angle into $\boldsymbol{e}^{(cab)}_\mathrm{CBF}$, and the dihedral angle into $\boldsymbol{e}^{(cabd)}_\mathrm{SBF}$. Using these vectors the atom and edge embeddings are updated in each layer via messages passed between neighboring atoms. Note that MPNNs use a fixed number of message passing steps, with separate learnable parameters in each step. This process starts with an initial atom embedding $\boldsymbol{h}^{(0)}_a = f_\mathrm{zemb}(z_a)$, based on the atomic number z_a, and an initial edge embedding $\boldsymbol{m}^{(0)}_{(ba)} = f_\mathrm{emb}(\boldsymbol{h}^{(0)}_b, \boldsymbol{h}^{(0)}_a, \boldsymbol{e}^{(ba)}_\mathrm{RBF})$. Figure S1 shows the full model architecture. For more details and the reasoning behind this model see Gasteiger, Becker, and Günnemann[4].

We trained the model using AMSGrad[6] with weight decay in combination with a linear learning rate warm-up, exponential decay and decay on plateau. We use the following

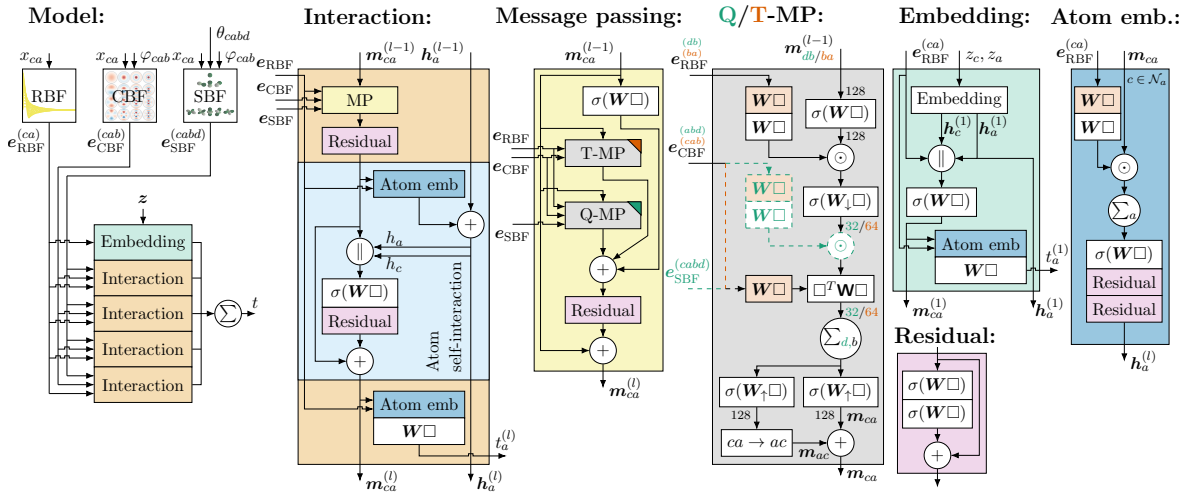


FIG. S1. The GemNet architecture[4]. □ denotes the layer's input, ‖ concatenation, σ a non-linearity (we use SiLU in this work[5]), and orange a layer with weights shared across interaction blocks. Differences between two-hop message passing (Q-MP) and one-hop message passing (T-MP) are denoted by dashed lines. Numbers next to connecting lines denote embedding sizes.

weighted loss function

$$\mathcal{L}_{\mathrm{MD}}(\boldsymbol{X}, \boldsymbol{z}) = (1-\rho)\left|f_\theta(\boldsymbol{X}, \boldsymbol{z}) - \hat{E}(\boldsymbol{X}, \boldsymbol{z})\right| + \frac{\rho}{N}\sum_{i=1}^{N}\sqrt{\sum_{\alpha=1}^{3}\left(-\frac{\partial f_\theta(\boldsymbol{X}, \boldsymbol{z})}{\partial \boldsymbol{x}_{i\alpha}} - \hat{F}_{i\alpha}(\boldsymbol{X}, \boldsymbol{z})\right)^2} \tag{1}$$

with the atomic coordinates $\boldsymbol{X}$, the force weighting factor $\rho = 0.999$, the model f_θ, target energy $\hat{E}$, target forces $\hat{F}$, and N atoms per molecule. The model implementation and training code is publicly available at `https://www.daml.in.tum.de/gemnet`.

C. Molecular dynamics simulations

All MD simulations discussed in the main text are performed in the NVT ensemble. MD simulations are propagated via the `atomic environment simulation package`[7] (ase) in combination with the Langevin thermostat and a custom GemNet calculator. We use a time step of 0.5 fs and a thermostat friction coefficient of 0.002 to ensure constant average temperatures of 300 K, 600 K and 1200 K. Mean absolute errors are evaluated for predicted atomization energies (AE) and force (F) components with the GemNet models and DFT reference calculations. To this end, 72 snapshots have been uniformly drawn from each MD

trajectory, after discarding a equilibration period of 100 ps.

To verify stability of the dynamics in the absence of a thermostat, we performed additional MD simulations for one molecule in the NVE ensemble using the 3.2Mio model (see Fig. S2). Here, 1 ns dynamics were performed with the Velocity Verlet algorithm as implemented in ase. A timestep of 0.5 fs was used and the velocities were initialized with a temperature of 1200 K from the Maxwell-Boltzmann distribution. It can be seen that the total energy is approximately conserved across the full simulation time. We further verified that the small fluctuations in the total energy are due to the finite timestep used, by reducing the timestep to 0.1 fs (see Fig. S3).

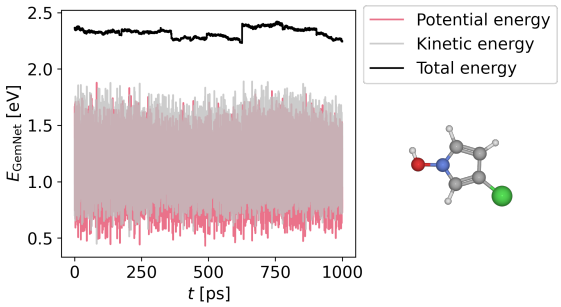


FIG. S2. Fluctuation of kinetic, potential and total energies for the shown molecule in the NVE ensemble. The potential energy calculated as the AE of the system minus the AE of the optimized geometry is shown in red. The kinetic energy is shown in gray and the total energy (potential energy + kinetic energy) in black.

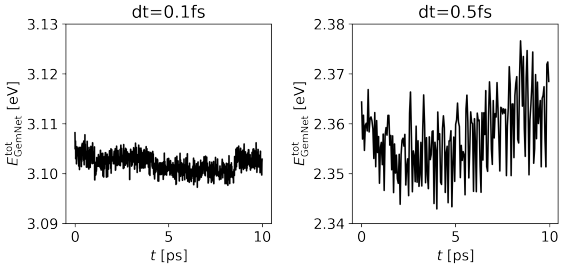


FIG. S3. Total energy as a function of simulation time for NVE MD simulations with a timestep (dt) of 0.1 fs (left) and 0.5 fs (right). Note that the different total energy scales are due to the independent initialization of both simulations leading to different starting temperatures.

II. LEARNING CURVE

The learning curves from Fig. 1 of the main text show mean absolute errors on AE and F as a function of training set size. To this end, a test set of 10,100 configurations was randomly drawn from the database. Unke and coworkers also use the same test set size in their studies on QM7-x.[8] The remaining data set was partitioned into sets of 4k, 40k, 400k and 4Mio configurations, while the smaller sets being included in the larger sets. Each of these sets underwent a further split, with 80% of the structures used for training and the remaining 20% for validation (i.e. to tune the hyperparameters of the models). In the end we thus obtain training set sizes of 3.2k, 32k, 320k and 3.2Mio configurations.

III. MOLECULAR DYNAMICS SIMULATIONS IN THE NVT ENSEMBLE

A. Exploration of configuration space

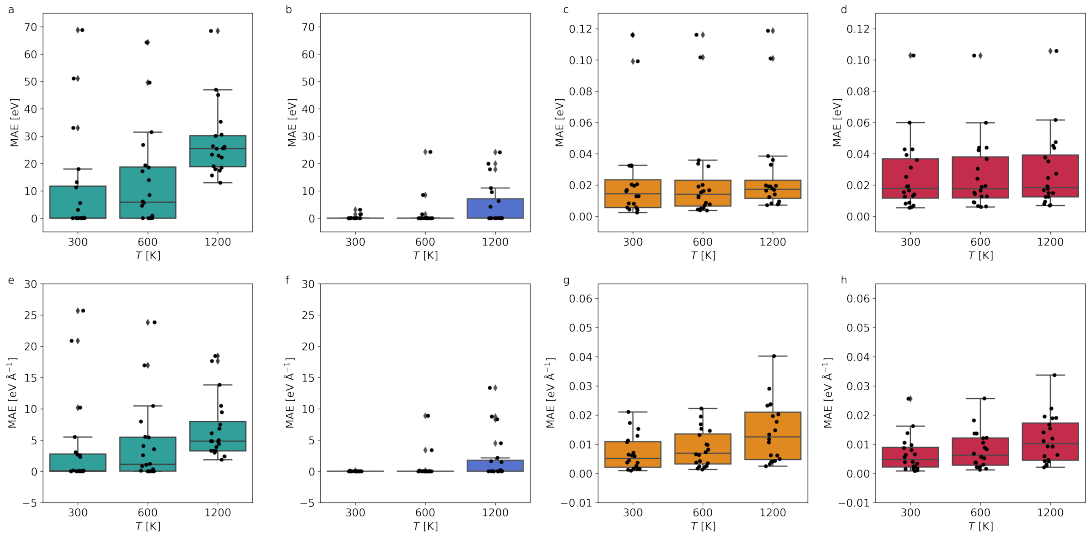


FIG. S4. Error distributions in MD trajectories. Results represented for molecules illustrated in the main text in Fig. 2a at various temperatures visualized as box plots.(a) AE MAE and (e) F MAE for the 3.2k model, (b) AE MAE and (f) F MAE for the 32k model, (c) AE MAE and (g) F MAE for the 320k model, (d) AE MAE and (h) F MAE for the 3.2Mio model. In addition, individual data points (black circles) are visualized.

As discussed in the main text, we test the robustness of various GemNet potentials by

running 1 ns MD trajectories for 20 different molecules at 300 K, 600 K and 1200 K in the NVT ensemble. While Fig. 2 in the main text showcases MAE trends of GemNet models trained on different training set sizes, here we additionally highlight the spread of mean absolute errors for both AE and F over the 20 molecules. Corresponding box plots are shown in Fig. S4.

B. Exploration of configuration and chemical space: Example Ser-Trp-Leu-tripeptide

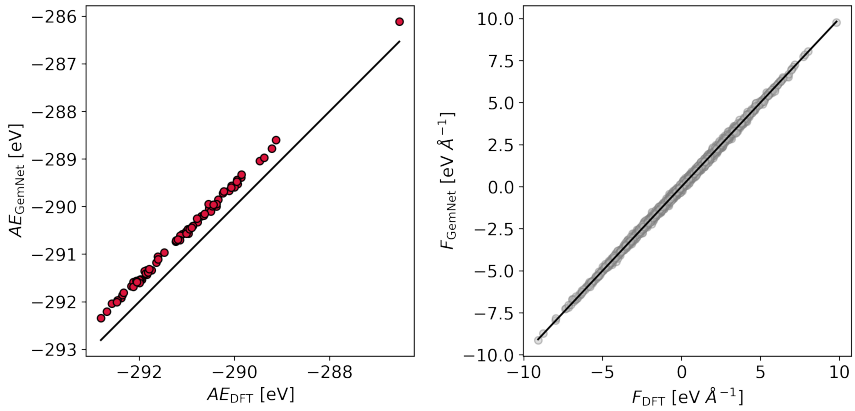


FIG. S5. Correlation plots for the large molecule (Ser-Trp-Leu-tripeptide). Left: AE, right: F components.

Figure S5 shows correlation plots between DFT calculated and GemNet predicted AE and F components for the Ser-Trp-Leu-tripeptide. We find an AE MAE of 0.47 eV and a F MAE of 0.02 eV Å^{-1}. The GemNet AEs are consistently shifted towards higher energies but highly correlated with the DFT reference values ($R^2 = 0.998$). This indicates that the GemNet error in this extrapolative regime is very systematic.

REFERENCES

[1]V. Blum, R. Gehrke, F. Hanke, P. Havu, V. Havu, X. Ren, K. Reuter, and M. Scheffler, Comput. Phys. Commun. **180**, 2175 (2009).

[2]J. Hoja, L. Medrano Sandonas, B. G. Ernst, A. Vazquez-Mayagoitia, R. A. DiStasio Jr., and A. Tkatchenko, Sci. Data **8**, 43 (2021).

[3]J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, in *ICML* (2017).

[4]J. Gasteiger, F. Becker, and S. Günnemann, in *NeurIPS* (2021).

[5]S. Elfwing, E. Uchibe, and K. Doya, Neural Netw. **107**, 3 (2018), special issue on deep reinforcement learning.

[6]S. J. Reddi, S. Kale, and S. Kumar, in *ICLR* (2018).

[7]A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. B. Jensen, J. Kermode, J. R. Kitchin, E. L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schtt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, J. Phys. Condens. Matter **29**, 273002 (2017).

[8]O. T. Unke, S. Chmiela, M. Gastegger, K. T. Schütt, H. E. Sauceda, and K.-R. Müller, "Spookynet: Learning force fields with electronic degrees of freedom and nonlocal effects," (2021), arXiv:2105.00304 [physics.chem-ph].