Ghent University

Optimizing Potential Energy Surface Models

Author:
Leonid Komissarov

Supervisor:
Prof. Dr. Toon Verstraelen

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Center for Molecular Modeling
Department of Physics and Astronomy

April 6, 2022

Preface

Before we begin I would like to take a few moments to explain the – to some certainly rather atypical – background of this doctorate. Atypical because it is not a classical doctorate, where PhD candidates spend most of their time at a research institution, accumulate their research results and present them in a thesis after a number of years. Instead, I was lucky to be part of an Industrial Doctorate, funded by the European Union's Marie Skłodowska-Curie Actions. The goal of such doctorates is to provide young researchers with experience and skills that go beyond academia. This is achieved through mobility, where fellows spend no more than fifty percent of their time at one beneficiary within their training network. Our network comprised three beneficiaries: Ghent University (Belgium), RWTH Aachen (Germany) and Software for Chemistry & Materials (Netherlands). Thanks to the fellowship, I had the opportunity to work on a variety of different topics with amazing colleagues across multiple countries and participate in a number of engaging training events.

This structure of the training network is reflected in the results of this thesis, as in addition to the academic publications, the main body of work has been the design and release of a software tool. I believe all parties largely benefit from such a project: Doctoral candidates from the additional experience, host institutions from the exchange, generation and retention of know-how, and the public from relevant and valuable research outcomes. At the end of my three years as an Early Stage Researcher, I am grateful to have had this unique opportunity and would like to thank everyone who made this possible.

First and foremost, I thank Toon for being the mastermind behind my project and a great supervisor. During my time he never said no to a single research idea of mine – no matter how far-fetched it was. All of his students are lucky to have him as a supervisor that is the personification of 'thinking outside the box' when it comes to combining ideas from multiple domains. I would like to thank Kai and Stan for making this project possible. A very special thanks to Sergio, who has been an organizational wizard of a project manager. Many thanks to all the ESRs on the project: Felix, Gabriel and Michael for sharing the same experience and being great colleagues.

For my time in Amsterdam, I would like to thank all of SCM for being an incredibly welcoming and friendly bunch. The fact that everyone is genuinely friendly and fun to talk to gives the company a special atmosphere. I would like to thank my office neighbors, Alexei and Matti, who always had the time to put up with my questions or idle chat. Thanks to Frieda for being the most caring office manager I have ever encountered – I would likely have had to sleep under a bridge in Ghent if it wasn't for her. Thanks to Hans for taking care of all things technical and system-administrative, especially those I could have solved myself after some thorough googling. Many thanks to Felix, Michal, Mirko, Peter-Pincers and Lars for joining me in the climbing gym close to eight times a week and the few beers afterwards. Thanks to Thomas, Mirko, Michal, Felix, Robert and of course Chef Michael for the fruitful and sometimes less fruitful political evenings. Many thanks to Robert, who contributed a great deal to ParAMS and always took care that my coding does not become too esoteric. Thanks to Lila, Clara and Nick for the best Monday evenings and for always welcoming us at their home – even when we practically invited ourselves.

For the time in Ghent, I would like to thank all members of the Center for Molecular Modeling for their hospitality, laid-back attitude and a lot of laughs at work and outside of it. Thanks to Michael and Tomi for helping me integrate into a new environment. A lot of thanks to Ruben, Massimo, Yingxing, Tomi and Michael for the numerous social events during what otherwise would have become a very lonely time. Thanks to Ruben, Klaas, Karen and Massimo for the climbing company. Thanks to Sander for all of the political jokes.

Many thanks to everyone who helped proofreading this thesis: Mirko, Matti, Fabi and Toon.

Lastly, thanks to my family for always having my back. Thanks to my dear friends from the school and university days for providing the necessary distraction from work. A special thanks to Joanna because she insisted on being mentioned by name.

Summary

The digital revolution has undoubtedly been a major contributor to the shaping of modern society. Nowadays computational simulations play an integral part in science and industry, enabling novel discoveries at an unprecedented pace. Such reliance on simulations means that there is a constant demand for hard- and software that produces results faster, more accurately and at a lower cost.

This thesis highlights a strategy that can deliver fast and accurate computational models: optimization; specifically in the context of empirical models of the potential energy surface, as used in physics and chemistry. Here, the problem of inaccurate predictions can be addressed by fitting model parameters to reference data. In doing so, a previously poor model can be trained to perform significantly better. Although this seems like a simple and viable approach on paper, the implementation is not straightforward: To date, there exists a plethora of models for molecular simulation, various optimization algorithms, and a number reference data sources. Until now the process of interfacing the above components has been tedious and prone to produce workflows that are of little comprehensive use to the scientific community.

We have developed a tool that alleviates these issues. It facilitates parameter optimization, allowing researchers to focus more on their science than the time-consuming technical details. In addition, the tool is highly flexible as users can mix and match various optimizers with different models and include any computable physicochemical property in the fitting procedure. The following pages provide an overview of how parametric molecular models can be optimized. We will discuss various types of models and optimization strategies before introducing our software tool. The advantages of our software are underlined with multiple successful parametrization examples. First applications of it resulted in improved performance of the ReaxFF and GFN1-xTB models. Both parametrizations have been made available to the scientific community and are discussed in the included papers.

Nederlandse Samenvatting

De digitale revolutie heeft ongetwijfeld een belangrijke bijdrage geleverd tot de vorming van de moderne samenleving. Tegenwoordig spelen computationele simulaties een integrale rol in wetenschap en industrie, waardoor in een ongekend tempo nieuwe ontdekkingen mogelijk worden. Deze afhankelijkheid van simulaties betekent dat er een constante vraag is naar hard- en software die resultaten sneller, nauwkeuriger en tegen lagere kosten oplevert.

Deze dissertatie belicht een strategie die snelle en nauwkeurige computationele modellen kan opleveren: optimalisatie; specifiek in de context van empirische modellen van het potentiële energie-oppervlak, zoals gebruikt in de natuur- en scheikunde. Het probleem van onnauwkeurige voorspellingen kan hiervoor aangepakt worden door modelparameters aan te passen aan referentiegegevens. Een initiëel slecht werkend model kan zo getraind worden om aanzienlijk beter te presteren. Hoewel dit op papier een eenvoudige en haalbare aanpak lijkt, is de uitvoering dat niet: Er bestaan een overvloed aan modellen voor moleculaire simulaties, verschillende optimalisatie-algoritmen, en een aantal bronnen voor referentiegegevens. Het proces van de integratie van de bovenstaande componenten was tot nu toe omslachtig en leidt tot workflows die van weinig waardevol zijn voor de wetenschappelijke gemeenschap.

Wij hebben een instrument ontwikkeld dat deze problemen verhelpt. Het vereenvoudigt de parameteroptimalisatie, waardoor onderzoekers zich meer kunnen ontfermen over wetenschappelijk aspecten dan op de tijdrovende technische details. Bovendien is het instrument erg flexibel, omdat gebruikers verschillende optimizers met verschillende modellen kunnen combineren en elke berekenbare fysisch-chemische eigenschap in de aanpassingsprocedure kunnen opnemen. Op de volgende pagina's wordt een overzicht gegeven van hoe parametrische moleculaire modellen kunnen worden geoptimaliseerd. We zullen verschillende soorten modellen en optimalisatiestrategieën bespreken alvorens ons softwareprogramma voor te stellen. De voordelen van onze software worden onderstreept met meerdere succesvolle parametriseringsvoorbeelden. De eerste toepassingen ervan resulteerden in verbeterde prestaties van de ReaxFF en GFN1-xTB modellen. Beide parametriseringen zijn beschikbaar gesteld aan de wetenschappelijke gemeenschap en worden besproken in de bijgevoegde papers.

Contents

To my grandparents

1 Introduction

Predictive models are omnipresent in our daily lives. We encounter them when looking up the weather forecast for the next weekend or in our email's spam filter. Mathematically, a model can be described as a function f that maps inputs $\boldsymbol{x}$ to outputs $\hat{\boldsymbol{y}}$:

$$f : \boldsymbol{x} \rightarrow \hat{\boldsymbol{y}}. \tag{1.1}$$

Consider a climate diagram that displays the distribution of temperatures throughout the year at one location. If a set of experimental data in the form of measurement dates $\boldsymbol{x}$ and temperatures $\boldsymbol{y}$ is available, we can use it as a reference to create a model that predicts temperatures $\hat{\boldsymbol{y}}$. Figure 1.1 shows an example of how such experimental data (discrete blue points) and a model (red line) could look like. To be able to judge whether the model is a good representation of the experimental data, a metric is needed. For this purpose a *loss function* $\mathcal{L}$ is commonly used, measuring the difference between the experimental and predicted points $\boldsymbol{y}$, $\hat{\boldsymbol{y}}$. One example of a loss function is the mean absolute error (MAE), defined as

$$\mathcal{L}_{\text{MAE}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|, \tag{1.2}$$

where N is the number of elements in the compared vectors. By design, larger loss function values translate to a poor model performance with respect to the reference data. In contrast, a model that could perfectly represent our temperature data from Figure 1.1 would have a loss of $\mathcal{L} = 0$. Since this is clearly not the case here, what can be done to improve the model? The model in Figure 1.1 is constructed from a second-degree polynomial, which has the form

$$f_2(x) = \sum_{i=0}^{2} p_i x^i = p_0 x^0 + p_1 x^1 + p_2 x^2. \tag{1.3}$$

It is *parametric* because in addition to the input x, a parameter vector $\boldsymbol{p} = (p_0, p_1, p_2)$ is needed to produce outputs. In such a case, we can optimize the values of $\boldsymbol{p}$ to better represent a set of reference data $\{\boldsymbol{x}, \boldsymbol{y}\}$.

Enabling and applying such optimizations is ultimately the goal of this thesis. A difference to the example is that the models of our interest are not (or at least not directly) related to the weather, but rather answer questions about

1

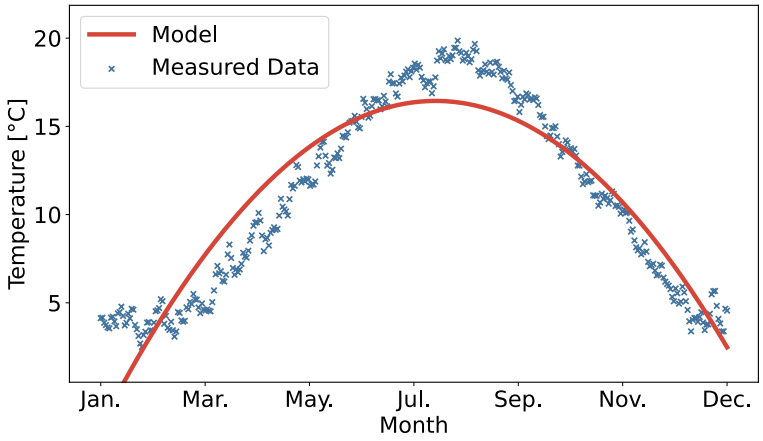


FIGURE 1.1: Example of a climate diagram showing experimental data [1] $\{x, y\}$ in blue, and a model fitted to that data in red.

the structure and properties of molecular systems. Models that predict chemical reactions, physical properties of a material, or the folded structure of a protein are perhaps not omnipresent in our daily lives, but equally important. After all, they allow researchers to investigate processes under conditions that are often beyond the scope of an experimental setup, or screen a large number of compounds as a means to select a few promising candidates for further evaluation. Novel polymers [2–4], fuels [5, 6], drugs [7–9] or batteries [10, 11] are examples of research outcomes that nowadays greatly rely on computational models.

In the following chapter, two relevant classes of molecular models are introduced and compared, which is necessary for a more accurate definition of this thesis' goals in Chapter 3. After a brief discussion of different model optimization approaches in Chapter 4, major research results are presented with the *ParAMS* software package in Section 5.1 and its application in Section 6.1. Section 6.2 introduces a reference data set, which can be used for future model development or benchmarking. Chapter 7 summarizes the major research results of this thesis and provides an outlook on future developments.

2 Modeling the Potential Energy Surface

This chapter provides an introduction to molecular models, commonly used to describe physicochemical systems in natural sciences. In particular, we are interested in the relationship between a system and its energy. A physical system is described by the coordinates of each individual particle in it. In the Cartesian space the configuration of a system of N particles is described by $3N$ coordinates:[1]

$$\boldsymbol{R} = (\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N) \in \mathbb{R}^{3N}. \tag{2.1}$$

Here, each particle's position is defined by its coordinates $\boldsymbol{r} = (r_x, r_y, r_z)$ and the subscript denotes its index. For example, a water molecule can be described as a system of three atoms, each with their own position. The *potential energy surface* (PES) can now be introduced as a hypersurface that assigns an energy value V to any possible configuration of $\boldsymbol{R}$. In analogy to Eq. 1.1, a model of the PES is a function that maps inputs of $\boldsymbol{R}$ to energy values:

$$f : \boldsymbol{R} \to V. \tag{2.2}$$

Note that although Eq. 2.1 is a complete description of a physical system for the calculation of the PES, other applications, such as thermodynamics require the addition of momenta. In addition to the energy, knowledge of the PES allows researchers to study many derived properties: We can, for example, calculate the forces acting on each atom; perform a geometry optimization of $\boldsymbol{R}$ by looking for minima on the PES; search for saddle points, which correspond to transition states in chemical reactions; perform a PES scan to selectively investigate an area of interest; run molecular dynamics (MD) simulations to investigate how a system behaves at a given temperature for a period of time, or calculate the vibrational frequencies of a system from the Hessian matrix.

As an example, Figure 2.1 depicts the PES of the $Cl^- + CH_3Br \rightleftharpoons Br^- + CH_3Cl$ reaction as a function of the C-Cl and C-Br distances. We can see that the system has two minima, each corresponding to the isolated molecules

[1] In the molecular case, atomic numbers are additionally needed to distinguish between different elements.

2

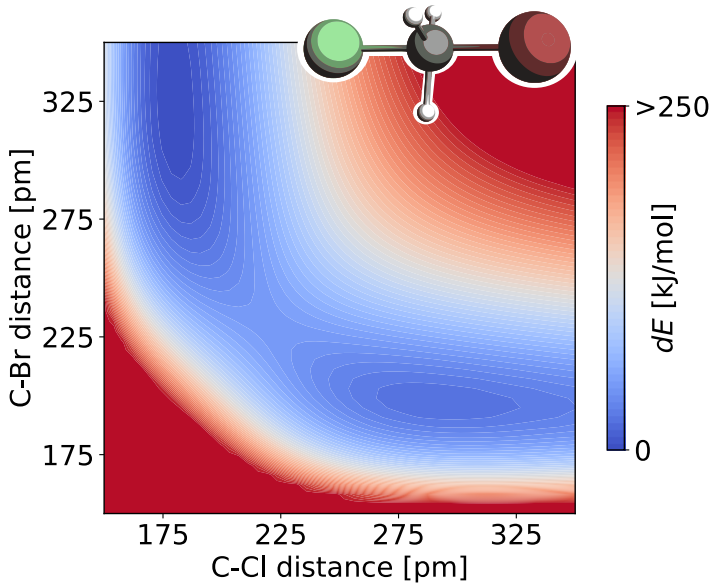


FIGURE 2.1: Slice of a PES for the $Cl^- + CH_3Br \rightleftharpoons Br^- + CH_3Cl$ system. Areas of low and high energy are depicted in blue and red, respectively. The atomic structure at the top right corresponds to the transition state at roughly $(225, 225)$. Chlorine, Bromide, Carbon and Hydrogen depicted in green, red, grey and white, respectively.

as per the chemical equation above. The system also has a transition state where both halogen atoms are roughly the same distance of 225 pm apart from the carbon atom. The corresponding structure is depicted in the upper right corner of the figure.

Naturally, the quality of any property that is derived from the PES highly depends on the quality of the underlying model. If we further consider the numerous research fields where simulations of the PES are applied, it is not surprising that at present there exists a myriad of different models, each tailored to specific applications. Here, we will classify models into two broad categories - *empirical* and *ab initio*. The following sections will discuss both.

2.1 Empirical Models

A model can be called empirical when it has been established *a posteriori*, after a number of observations and its general validity is limited to a specific domain. The weather example from Figure 1.1 can be considered such a case: It is only possible to accurately model the observed data after many temperature measurements throughout the year. The model also does not transfer well to temperature predictions at other locations across the globe. Put differently, the construction of an empirical model does not require us to understand *why* a process is happening – all we need is to observe it. Empirical development of molecular models, where interactions between multiple

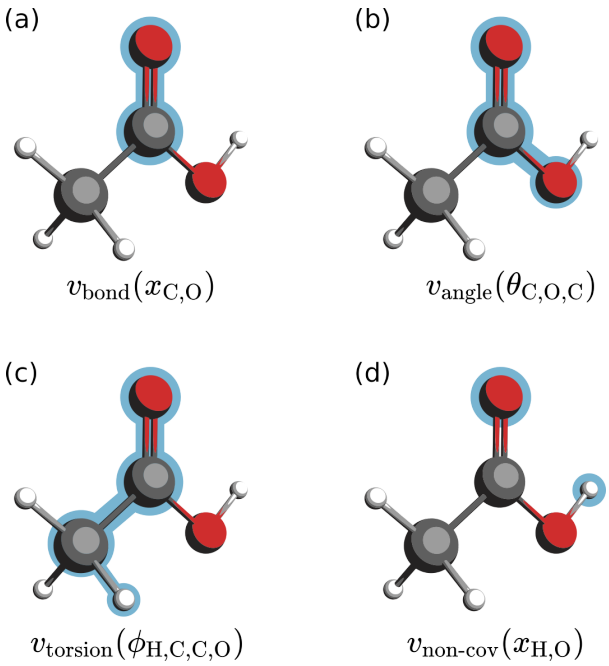


FIGURE 2.2: Examples of how the individual potentials of a force field describe distinct features of a chemical system. Depicting bond distances, atomic angles, torsion angles and non-covalent interactions in (a) to (d), respectively. Blue highlights mark the atoms considered for the calculation of each potential, as also represented by the subscript of each potential's input variable. Carbon, hydrogen and oxygen atoms are shown in grey, white and red, respectively.

particles are described based on previous observations, is no exception to this.

The family of classical force fields (FFs) is by far the most prominent example of empirical descriptions of the PES. On an atomistic scale, classical force fields only consider interactions between nuclei, without explicit treatment of the electrons. The PES of a generic force field is usually a sum of multiple contributions to the total energy:

$$V_{\text{FF}} = V_{\text{bonds}} + V_{\text{angles}} + V_{\text{torsions}} + V_{\text{non-covalent}}, \tag{2.3}$$

where the terms on the right hand side denote individual potentials for bond stretching, angle bending, torsion angle rotation and non-covalent interactions, respectively. Graphical examples of the features described by each potential are provided in Figure 2.2, depicting how each targets a different geometrical property in the same system. The potentials are functions of multiple atomic coordinates, as highlighted in blue and denoted by the subscript of each potential's input variable.

Technically, force field developers are free to implement any arbitrary equation for a potential. Typical choices for classical potentials are based on chemical intuition and will oftentimes include harmonic potentials of the form

$$v(x, x_0, k) = \frac{1}{2}k(x - x_0)^2 \tag{2.4}$$

to describe bond stretching and angle displacement. Here, x is an internal coordinate (*e.g.* bond distance or angle), x_0 is the equilibrium position of that coordinate and k a force constant. The latter two are model parameters. Energy fluctuations due to torsion are usually modeled by periodic functions. The Lennard-Jones [12, 13] and Coulomb potentials are common choices to describe van-der-Waals and electrostatic interactions, respectively.

A drawback of classical force fields is their inability to describe chemical reactions. In fact, many force fields explicitly require that users provide a fixed bonding table in addition to the coordinates, which can be an issue for systems where the number of bonds can not be determined easily. The ReaxFF reactive force field formalism [14] addresses this shortcoming by introducing a more complex potential. Here, the definition of continuous bond orders allows ReaxFF to describe the formation and breaking of bonds. Due to this feature the model has seen a wide range of applications. We will discuss the parametrization of ReaxFF in Section 5.1. For a good overview of the method and its applications, see Ref. 15.

The empirical approach is taken one step further in the fairly recent class of machine-learning potentials (MLPs). One common way to construct MLPs is through artificial neural networks (NNs) [16–18]. The core building block of NNs is the neuron, which transforms an n-dimensional input vector $\boldsymbol{x}$ to a scalar output y as follows:

$$y = f(\boldsymbol{w} \cdot \boldsymbol{x} + b) = f\Big(\sum_i^n w_i x_i + b\Big), \tag{2.5}$$

where $(\cdot)$ is the dot product, $\boldsymbol{w}$ is a weights vector with the same dimensionality as $\boldsymbol{x}$, b is a scalar bias, and f is an activation function. Here, $\boldsymbol{w}$ and b are model parameters. Additional model complexity is added through the stacking of multiple neurons, producing a neural network layer. This is visualized in Figure 2.3, showing how the dimensionality of the output y in Eq. 2.5 can be expanded to an arbitrary output vector of m dimensions. The stacking is conceptually straight-forward, transitioning from an n-dimensional weights vector $\boldsymbol{w}$ to a weight matrix $\boldsymbol{W}$ of $m \times n$ elements and an m-dimensional bias vector $\boldsymbol{b}$. Similar to the stacking of neurons into a layer, multiple layers can be chained by using the output of a layer $(n-1)$ as the input to a consecutive layer n. This allows the design of NNs with an arbitrary number of layers,

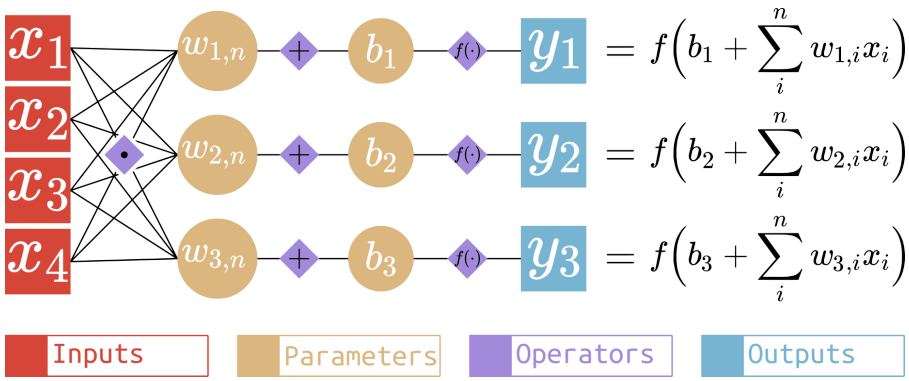


FIGURE 2.3: A schematic representation of a one-layer neural network. The n-dimensional input vector $\boldsymbol{x}$ (red) is transformed to a m-dimensional output vector $\boldsymbol{y}$ (blue) through $\boldsymbol{y} = f(\boldsymbol{W} \cdot \boldsymbol{x} + \boldsymbol{b})$, where $\boldsymbol{W}$ is a $m \times n$ weights matrix, $\boldsymbol{b}$ the bias vector and f is a activation function. Here, $n = 4$ and $m = 3$, making for a total of 12 parameters in $\boldsymbol{W}$ and an additional 3 in $\boldsymbol{b}$.

where each layer can contain an arbitrary amount of neurons. Unlike with classical force fields, where the functional form of the PES is predetermined by design (*e.g.* the shape of Eq. 2.4), neural networks, in theory, can approximate any shape of data in the Euclidean space. This property is referred to as the universal approximation theorem [19]. In the case of MLPs the final output at the end of a network is typically the total system energy, as a sum of all atomic energies. Initial inputs are a transformation of $\boldsymbol{R}$, for example through symmetry functions [16,17,20] or in the form of a graph representation [21–23].

Due to their fairly low computational cost and an increasing availability of scalable hardware in the recent years, empirical models are an attractive choice for the simulation of long time-scales or large systems up to millions of atoms [24–27]. Typical applications include molecular docking [28,29] or the studies of bulk-phase material properties [24,30]. The downside of such models is a limited chemical space in which predictions are accurate, as will be discussed in Section 2.3.

2.2 Ab Initio Models

In direct contrast to the empirical approach, the underlying concept of *ab initio* models can be traced back to the question of *why* a process is happening. Their naming (Latin 'from the beginning') is due to such models being constructed from the fundamental laws of physics only. For molecular models, this encompasses the explicit treatment of all electrons and nuclei within a system.

To account for the fact that the microscopic states of a system are quantized, meaning that some observables take discrete rather than continuous values,

ab initio methods are based on quantum mechanics (QM). Here, any physical system is described by a wave function $\Psi(\boldsymbol{R})$. A central piece for the ab initio modeling of the PES is the time-independent Schrödinger equation [31], which describes the energy E of a QM system by

$$\hat{H}\Psi(\boldsymbol{R}) = E\Psi(\boldsymbol{R}) \; \forall \; \boldsymbol{R} \in \mathbb{R}^{3N}, \tag{2.6}$$

where $\hat{H}$ is the Hamiltonian.

Unfortunately, finding an exact solution to the above equation is not easily possible for any but the simplest of systems, as iterative approaches quickly become out of reach due to the exponential scaling of the problem. There are, however, approximations that nevertheless allow for the construction of an accurate PES at a reasonable computational cost. Density functional theory (DFT) provides one of many approximate solutions to the Schrödinger equation and is a popular choice for accurate computations of relatively large systems. The following will discuss DFT assuming the Born-Oppenheimer approximation [32], stating that the movements of nuclei and electrons can be separated from each other due to their large difference in masses. Practically, this means that for any system of interest $\boldsymbol{R}$, only the positions of the electrons are considered, with N being the total number of electrons. Formally, the spin is an additional parameter of the wave function when treating electrons explicitly. However, because the following quantities do not depend on the spin (after integration), we will ignore it in favor of a clear notation.

Density functional theory

A fundamental concept of DFT is the replacement of the many-body wave function $\Psi(\boldsymbol{R})$ with one that can be constructed from the electron density $\rho(\boldsymbol{r})$ of a system [33]

$$\rho(\boldsymbol{r}) = N \int \cdots \int |\Psi(\boldsymbol{r}, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N)|^2 \, d\boldsymbol{r}_2 \ldots d\boldsymbol{r}_N \; \forall \; \boldsymbol{r} \in \mathbb{R}^3. \tag{2.7}$$

Following Kohn and Sham [34], the ground state energy of a system is determined iteratively by constructing the electron density from a set of single-electron functions $\{\varphi\}$:

$$\rho = \sum_i^N |\varphi_i|^2, \tag{2.8}$$

and minimizing the total energy

$$E(\rho) = T(\{\varphi\}) + \int \rho(\boldsymbol{r}) v_{\text{ext}}(\boldsymbol{r}) \, d\boldsymbol{r} + E_{\text{coul}}(\rho) + E_{\text{xc}}(\rho). \tag{2.9}$$

The equation describes kinetic energy T of a fictional system of non-interacting fermions in an effective potential. The latter accounts for the energy due to nuclei-electrons, repulsive coulomb, and exchange-correlation interactions, described by the second, third and fourth terms in the above equation, respectively. While the first three terms in the above equation are exact, only approximations to E_{xc} exist. A number of suitable exchange-correlation functionals are available to date, generally differing in their accuracy and computational cost. From a computational efficiency point of view, DFT calculations scale in the order of $N \log N$ to N^3 [35, 36], making it an attractive method for quantum chemical calculations of moderate system sizes up to hundreds of atoms.

The semi-empirical density-functional tight-binding (DFTB) method exists as an approximation to DFT. It describes the electronic energy of a system as a Taylor expansion around a reference density ρ_0 and its fluctuations $\delta\rho$, such that $\rho = \rho_0 + \delta\rho$ [37–39]. DFTB makes use of minimal basis sets and element-specific parameters for the construction of atomic orbitals. As the reference density does not change in DFTB, the simplest models can compute an energy without the need for the computationally expensive self-consistent minimization procedure that is necessary in DFT. We will present a parametrization of a DFTB model in Section 6.1.

Note that aside from DFT, wave function based methods greatly contribute to the world of ab initio models. Although not discussed here, the interested reader is referred to References 36, 40, 41.

2.3 Model Applicability

Putting specific examples for PES models aside, this section provides a bird's-eye-view of the two model types with respect to their performance. Specifically, we will discuss the computational speed and the chemical space that a model can cover.

Given a system of size N (atoms in the case of classical models; electrons in the case of ab initio models), computation times of empirical models scale at worst with N^2, whereas ab initio models typically scale anywhere from N^3 to N^7 [41, 42]. For empirical models, this means that the doubling of a system size will result in a four times longer calculation. In the case of ab initio models, a calculation with a two times larger system will take anywhere from eight to 128 times longer.

At this point we can introduce the concept of the effective chemical space, as a means to assess the broader applicability of a model. The chemical

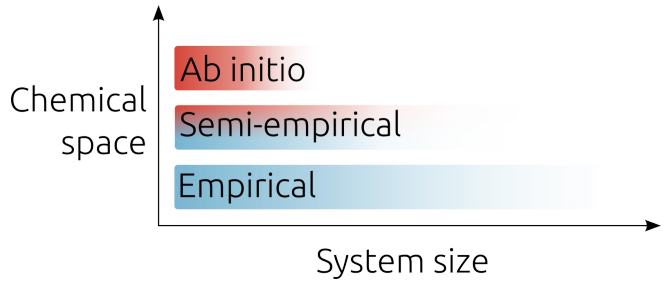


FIGURE 2.4: A generalized classification of empirical (blue), semi-empirical (red-blue) and ab initio (red) models based on the effective system size and chemical space coverage. Empirical methods can simulate larger system sizes due to favourable scaling, but are otherwise only accurate within a relatively small chemical space. In contrast, ab initio models are accurate throughout a larger space, but are limited in the overall system size they can simulate. Semi-empirical models try to bridge the gap by following the approach of ab initio models, but introduce approximations that simplify the calculation of the PES (see Ref. 36 for more details).

space is a theoretical construct that describes an ensemble of chemical systems, arranged by their properties, for example functional groups or polarity. Considering up to 30 atoms and only the elements C, H, O, N, S, conservative estimates predict that theoretically, at least 10^{63} unique, chemically valid compounds can be constructed [43]. Given a chemical space, a PES model's coverage of it entails the subspace of systems for which computational results can be computed with a certain level of accuracy.

Considering the ensemble of all ab initio models, their major advantage is that they are not limited to any particular subspace, meaning that one can expect to achieve the same quality of results regardless of the input system (method-specific limitations of individual ab initio models do exist, but we ignore these for the sake of a generalized comparison to empirical models). However, due to their unfavorable scaling, calculations of systems larger than a few hundreds of atoms become infeasible as computation times quickly soar past the average life expectancy of a human. For this reason simulations that involve large systems are predominantly carried out with empirical models. Although system size is not a major issue with empirical models, their covered chemical space is nevertheless comparably small due to deliberate simplifications. In practice, this can result in the presence of different parameter sets for one model, each set aimed at a specific application such as combustion or condensed phase simulations [15].

Figure 2.4 summarizes the above comparison, assigning each model class a position on a two-dimensional graph. Here, we implicitly include the model accuracy in the chemical space descriptor. This is because without a specific application, the measure of accuracy is difficult to quantify. Matching experimental results, the term *chemical accuracy* is often used when considering errors in the prediction of thermochemical energies with a value of 1 kcal/mol

or lower [42]. For other properties, however, such a definition might not be as straight-forward for multiple reasons, like the lack of a well-defined experiment. Note that the classification of a model as empirical or ab initio is not always binary: Many models fall into the so-called *semi-empirical* class, meaning that parts of it are empirical, while others are not. Performance of such models is usually situated between the two classes discussed here. The broad set of ab initio models is nowadays regarded as the most accurate class of PES models for many applications. For an overview of the benchmarking of ab initio models, refer to Reference 44.

3 Motivation and Goals

The overarching motivation for this thesis is the ambition to create a PES model that is both accurate and fast, saving researchers' time and businesses' money. Since, as discussed previously, a "chemical model of everything" does not exist, other strategies need to be pursued. The approach followed here is the fitting of empirical models to reference data. This acknowledges that the validity of empirical models is limited to a small chemical space, but assumes that they interpolate well within a region. It is thus possible to create fast and accurate models constrained to specific applications.

To demonstrate this, let us revisit the weather prediction example from Chapter 1. Here, the twelve months of the year are an analogy to the complete chemical space. Figure 3.1a shows the already familiar diagram with a quadratic model (red curve) fitted to all discrete points in blue. Arguably, the model is not representing the fitted data very well, especially in the months of January to May. This can be corrected by limiting the fit to a smaller region, as shown in Figure 3.1b, where the same model was fitted to the first five months only. As a result, the new model is more accurate within the smaller region, but at the same time entirely fails to describe the latter part of the year.

The same concept can be applied to models of the potential energy surface, where months of the year are replaced with molecular structures and temperatures with physicochemical properties. Despite the straight-forward idea, a number of hurdles make the task difficult in practice. Most of these are owed to the lack of flexible, standardized data formats. As such, the first goal of this thesis is to introduce a tool that makes optimization of chemical models both accessible to non-expert users and attractive to advanced researchers alike. The former group of users expects an easy-to-use tool that can deliver improved results with minimal effort, while the latter is usually interested in additional functionality such as advanced data set composition and parameter analysis. The tool should support (a) flexible definition of reference data from several origins, usually from experimental results or ab initio calculations, (b) multiple optimization algorithms and (c) a variety of empirical models that can be fitted to the reference data. Extension of any of the three components to a larger pool of algorithms should be straightforward and will guarantee that the tool will continue to be relevant in the future (d).

3

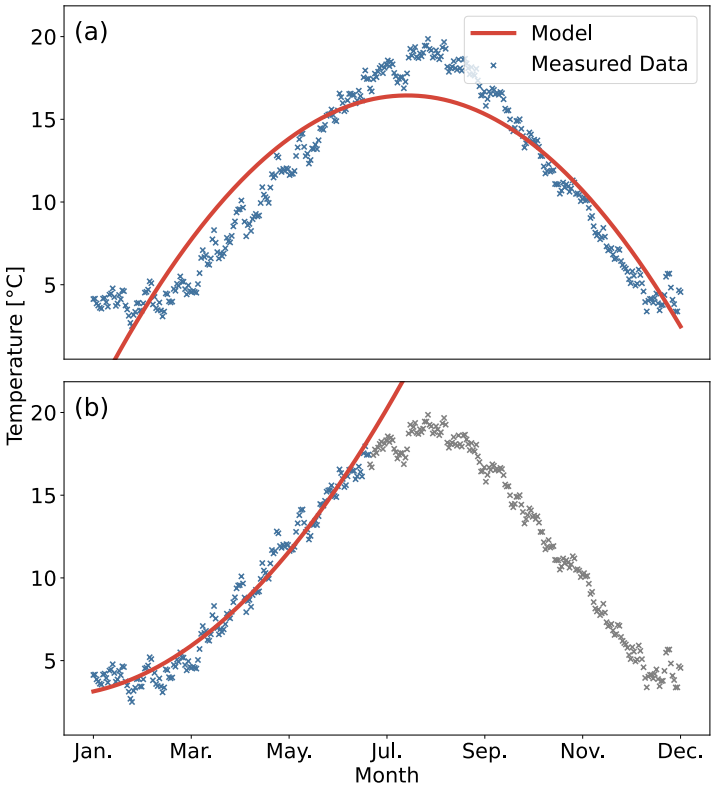


FIGURE 3.1: Example of a climate diagram showing experimental data [1] as discrete points, and a second-degree polynomial fit to the data as a red line. All discrete points were used for the fit in (a), whereas only the first five months were used in (b).

After discussing approaches to parameter optimization in the following Chapter 4, the proposed package for such tasks – named *ParAMS* – is introduced in PAPER I (*cf.* Section 5.1). Chapter 5 highlights how ParAMS integrates with already available molecular simulation packages. In the same paper, two first applications are presented: A parametrization of the ReaxFF reactive force field [14] for disulfides, and a second for the repulsive potential of DFTB [38] for zinc oxide.

Chapter 6 presents two additional publications related to the optimization of molecular models: In PAPER II we apply our package to a larger parametrization workflow for the GFN1-xTB [45] model. A standalone reference data set for zeolite structures is presented in PAPER III before moving on to the conclusion.

Finally, the results are summarized in Chapter 7, which also provides an outlook into possible future developments.

4 Parameter Optimization

How do we create a model that is both fast and accurate? In the world of empirical models, parameter optimization is one answer to that question. From the discussion in Section 2.1, we know that the functional form of an empirical model can be adjusted through a set of parameters. Practically, this means that the parametric model is not only a function of the input data $\boldsymbol{x}$, but also of the parameter vector $\boldsymbol{p}$. If an optimization of parameters is performed given a static data set $\{\boldsymbol{x}, \boldsymbol{y}\}$, the parameter vector will be the *only* variable that influences a model's functional shape. This can be written as

$$f(\boldsymbol{p}|\boldsymbol{x}) = \hat{\boldsymbol{y}}, \tag{4.1}$$

reading as 'function of $\boldsymbol{p}$, given $\boldsymbol{x}$'. A measure in the form of a loss function $\mathcal{L}$ is needed to express how accurate the outputs $\hat{\boldsymbol{y}}$ of a model are as compared to some reference data $\boldsymbol{y}$. An optimal parameter set $\boldsymbol{p}^*$ can then be defined as the one that minimizes the loss function:

$$\boldsymbol{p}^* = \min_{\boldsymbol{p}} \mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{y}}), \tag{4.2}$$

where, once again assuming that the training data does not change, the loss function can be expressed solely as a function of the parameter vector

$$\mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{y}}) \equiv \mathcal{L}\Big(\boldsymbol{y}, f(\boldsymbol{p}|\boldsymbol{x})\Big) \equiv \mathcal{L}(\boldsymbol{p}|\boldsymbol{x}, \boldsymbol{y}, f) \tag{4.3}$$

in the context of parameter optimization. Different approaches to finding $\boldsymbol{p}^*$ are discussed in the following sections.

4.1 Gradient Methods

The point $\boldsymbol{p}^*$ is a local minimum of a function $\mathcal{L}$ if the two conditions

$$\nabla\mathcal{L}(\boldsymbol{p}^*) = 0 \tag{4.4}$$

$$\boldsymbol{z}^{\mathrm{T}}\mathbf{H}_{\mathcal{L}}(\boldsymbol{p}^*)\boldsymbol{z} \geq 0 \;\forall\; \boldsymbol{z} \in \mathbb{R}^n, \tag{4.5}$$

are met. Here, $\nabla\mathcal{L}$ and $\mathbf{H}_{\mathcal{L}}$ are the gradient and Hessian of $\mathcal{L}$. The family of gradient-based optimization algorithms is exploiting at least one of the above

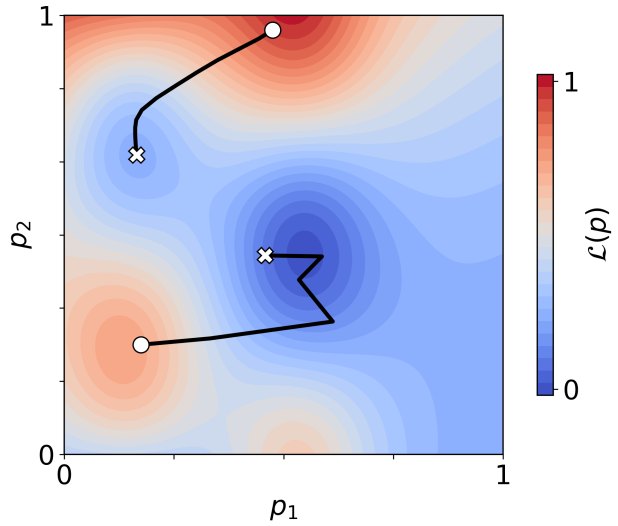


FIGURE 4.1: Two *gradient descent* optimization trajectories (black lines) with starting and end points marked with a circle and cross, respectively. The lower trajectory does not converge at the local minimum due to a too large step size. Blue and red regions mark low and high loss function values, respectively.

equations to arrive at a minimum. A simple representative is the *gradient descent* method (also called *steepest descent*) which, starting from a point $\boldsymbol{p}_i$ iteratively calculates the next point through

$$\boldsymbol{p}_{i+1} = \boldsymbol{p}_i - \alpha \nabla \mathcal{L}(\boldsymbol{p}_i), \tag{4.6}$$

where $\alpha > 0$ is the step size parameter. The algorithm can be described as taking a step α in the direction of the steepest descent, as defined by the negative gradient of $\mathcal{L}$.

Black curves in Figure 4.1 display two gradient descent optimization paths on a two-dimensional function, with start and end points marked with a circle and a cross, respectively. We can observe two shortcomings of the algorithm in the figure: As a consequence of the algorithm's local character, we see that the optimal solution is highly dependent on the initial position. Furthermore, the common issue of a too large step size α can be observed in the lower optimization trajectory. Here, the optimization never converges as the step size causes the next guess to go past the local minimum. In contrast, step sizes that are too small often result in a slow convergence. Such issues are usually addressed with an adaptive α [46–48]. Other algorithms that compute a *Newton step* require the calculation of a Hessian $\mathbf{H}_{\mathcal{L}}$ (or an approximation thereof) [49].

4.2 Derivative-Free Methods

In contrast to the above, the optimization of functions for which derivatives are unavailable or uninformative is carried out with derivative-free optimization (DFO) algorithms. Examples for such cases is the optimization of experimental setups, the tuning of hyperparameters, or in cases where a black box software is behind the evaluation of $\mathcal{L}$. Several approaches to DFO exist: (a) finite differences can be used to approximate the derivatives, (b) direct search methods sample each coordinate direction individually to arrive at a lower point at each iteration, (c) trust region methods generate an approximate model based on the current best solution and use it to determine the next step [49, 50].

Unfortunately, few of these approaches are satisfactory when searching for a minimum on a high-dimensional, noisy and discontinuous loss function, as will be depicted in the following section. Instead, metaheuristic methods such as simulated annealing [51, 52] or evolutionary algorithms [53, 54] are used to sample difficult spaces and find an optimal solution. Such methods are often disregarded by the numerical optimization community, as they lack a rigorous convergence theory [50]. Nevertheless among these, the covariance matrix adaptation evolution-strategy (CMA-ES) [55,56], developed by Hansen and Ostermeier has seen a broad application in many fields of research where conventional optimization fails [57–60]. As we use the CMA algorithm in Paper I (Section 5.1) and Paper II (Section 6.1), a detailed description of it follows.

Covariance matrix adaptation evolution-strategy

CMA-ES is an evolutionary algorithm which works with a population of candidate solutions at every iteration. The population is drawn pseudo-randomly from a normal distribution with a covariance matrix. After evaluation of all solutions, a subset with the lowest loss function values is used to update the covariance matrix for the next iteration. Combined with heuristics for the sampling width and learning rates, this approach is expected to explore regions with consecutively lower loss function values.

The algorithm samples $\lambda \in \mathbb{Z}_{>1}$ solution candidate vectors from an n-dimensional multivariate normal distribution $\mathcal{N}$ at each iteration g such that

$$\{\boldsymbol{p}^{(g+1)}\}_\lambda \equiv \boldsymbol{p}^{(g+1)}_{1,\dots,\lambda} \sim \sigma^{(g)} \mathcal{N}(\bar{\boldsymbol{p}}^{(g)}, \boldsymbol{C}^{(g)}), \tag{4.7}$$

where $\sigma \in \mathbb{R}_{>0}$ is the step size, $\bar{\boldsymbol{p}} \in \mathbb{R}^n$ and $\boldsymbol{C} \in \mathbb{R}^{n \times n}$ are the learned mean solution and covariance matrix, respectively. After evaluation of all parameter

candidates on $\mathcal{L}$, the mean $\bar{\boldsymbol{p}}$ is adjusted to the weighted average of

$$\bar{\boldsymbol{p}}^{(g+1)} = \sum_{i}^{\mu} w_i \boldsymbol{p}_i^{(g+1)}, \tag{4.8}$$

assuming that all $\{\boldsymbol{p}^{(g+1)}\}_\lambda$ have been ranked by their function values $\{\mathcal{L}(\boldsymbol{p})\}_\lambda$ (low to high). $\mu \leq \lambda$ is the number of candidates that are considered for the update, and $\boldsymbol{w} \in \mathbb{R}^{\mu}_{>0}$ is the recombination weights vector with $\sum w_i = 1$. The covariance matrix, set to the identity matrix at the beginning of the optimization, is updated as follows:

$$\boldsymbol{C}^{(g+1)} = (1 - c_\mu - c_1)\boldsymbol{C}^{(g)} + c_1 \, \boldsymbol{a}_c^{(g+1)} \boldsymbol{a}_c^{(g+1)^{\mathrm{T}}} + c_\mu \, \boldsymbol{C}_\mu^{(g+1)}. \tag{4.9}$$

The second and third terms of Eq. 4.9 are called rank-one and rank-μ updates respectively with positive learning rates $c_1 + c_\mu \leq 1$. $\boldsymbol{a}_c$ is the covariance matrix evolution path, calculated as

$$\boldsymbol{a}_c^{(g+1)} = (1 - c_c)\boldsymbol{a}_c^{(g)} + \sqrt{\frac{(2 - c_c)c_c}{\sum w_i^2}} \, \frac{\bar{\boldsymbol{p}}^{(g+1)} - \bar{\boldsymbol{p}}^{(g)}}{\sigma^{(g)}}, \tag{4.10}$$

and describes the mean solution's normalized trajectory, where the parameter $c_c \leq 1$ is the path learning rate. As the rank-one update retains information from previous iterations, it can be interpreted as the long term memory of the update. In contrast, the rank-μ update covariance matrix is calculated from all μ candidate vectors but does not accumulate any information regarding previous iterations:

$$\boldsymbol{C}_\mu^{(g+1)} = \frac{1}{{\sigma^{(g)}}^2} \, \sum_{i}^{\mu} w_i \big(\boldsymbol{p}_i^{(g+1)} - \bar{\boldsymbol{p}}^{(g)}\big)\big(\boldsymbol{p}_i^{(g+1)} - \bar{\boldsymbol{p}}^{(g)}\big)^T. \tag{4.11}$$

Combination of both updates into Eq. 4.9 increases the likelihood of successive candidates being sampled from a 'valid' space with respect to parameter co-dependencies. The step size σ is updated in a similar way to Eq. 4.10, following an evolution path $\boldsymbol{a}_\sigma$ such that

$$\sigma^{(g+1)} = \sigma^{(g)} \exp\Big(\frac{||\boldsymbol{a}_\sigma^{(g+1)}||_2}{\sqrt{n}} - 1\Big), \tag{4.12}$$

where $||\cdot||_2$ is the Euclidean norm. For a more detailed introduction and background to the CMA-ES algorithm, the tutorial in Ref. 61 is highly recommended.

Figure 4.2 shows three snapshots from a CMA-ES optimization, with 4.2a, 4.2b and 4.2c showing iterations number one, three and seven, respectively. At each iteration the sampled population size of $\lambda = 20$ is depicted as X

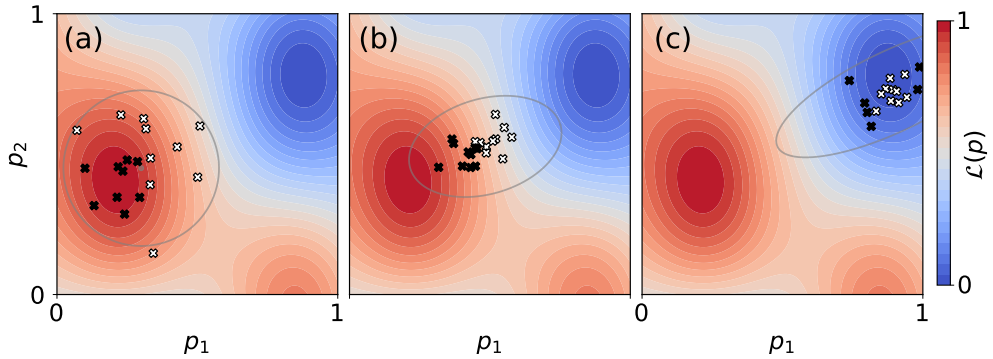


FIGURE 4.2: Three snapshots from a CMA-ES optimization trajectory at iteration one, three and seven in (a) to (c), respectively. The population $\lambda = 20$ is plotted as X marks. Points in white mark the parent subset $\mu = 10$. Error ellipsoid of the covariance $\boldsymbol{C}$ and mean solution $\bar{\boldsymbol{p}}$ are depicted in grey. Blue and red regions mark low and high loss function values, respectively.

marks. Out of the complete population, the parent subset of $\mu = 10$ is marked in white. These values are used to update the covariance matrix $\boldsymbol{C}$. Its error ellipse and the mean $\bar{\boldsymbol{p}}$ are depicted in grey. Points with too high loss function values are discarded, as marked in black.

4.3 Search Spaces

When optimizing PES model parameters, a general notion about the shape of the parameter search space can be very valuable when it comes to the choice of appropriate optimization algorithms and the expected outcome of a parameterization. Just like the PES, the parameter search space can be described by a high-dimensional surface where every point $\boldsymbol{p}$ is assigned a loss function value $\mathcal{L}(\boldsymbol{p})$. Its dimensionality is directly determined by the number of model parameters that are optimized.

To provide a perspective of the dimensionality, consider a model constructed from interatomic Lennard-Jones potentials [12,13] that have the form

$$v_{\mathrm{LJ}}(r, \boldsymbol{p}) = \frac{p_1}{r^{12}} - \frac{p_2}{r^6}, \tag{4.13}$$

where r is the interatomic distance and $\boldsymbol{p}$ is a parameter vector. Since this is a pairwise potential (the distance r is measured between two atoms), a model of n chemical elements would require $\frac{n!}{2(n-2)!} + n$ potentials to describe all possible bonds and twice as many parameters (two per potential). Such combinatorics can make the optimization problem difficult as the search space rapidly increases with the dimensionality of the parameter vector $\boldsymbol{p}$.

In the case of neural network potentials, the number of parameters scales with the number of hidden layers and their output dimensionality (*cf.* Eq. 2.5). Typically, force fields contain in the range of 100 to 10^3 parameters, [14,62,63]

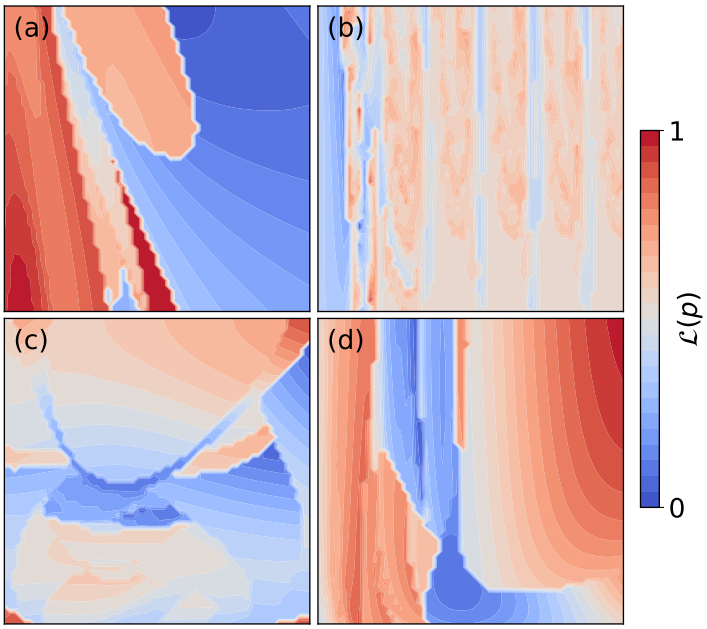


FIGURE 4.3: Four slices through a high-dimensional parameter search space of a ReaxFF reactive force field model [14]. Spaces can be bordered by restricted regions (a, d), noisy (b) or contain multiple minima (b,c,d), making optimization difficult (and looking like abstract art). Blue and red regions mark low and high loss function values, respectively. Loss values are normalized.

while up to 10^7 parameters [17, 64] are common for NNs. When comparing two similar models, the one with the larger number of parameters will be able to cover a larger chemical space but in exchange, parameter optimization in a higher dimensionality will be slower.

When optimizing molecular models, the parameter search space is often non-smooth, discontinuous or noisy, which is the main reason many conventional optimization algorithms fail and stochastic approaches are needed. Examples of how parameter search spaces can look like are provided in Figure 4.3. Each of the shown surfaces was computed by scanning parameters from the ReaxFF [14] force field model (*cf.* PAPER I), where red and blue represent regions of high and low loss function values, respectively. All four examples show surfaces that are difficult to optimize due to their topologies.

From Eq. 4.3, we see that there are two main variables that determine the shape of the loss function: The parametric model f, and the composition of the training data set $\{\boldsymbol{x}, \boldsymbol{y}\}$. This is critical when dealing with difficult parametrization tasks. Users should carefully consider the composition of a training set as this can translate into faster or slower optimization convergence and the overall quality of the obtained parameters.

5 Software

ParAMS – the optimization package developed during the course of this doctorate (*cf.* Section 5.1) – is integrated within the Amsterdam Modeling Suite (AMS) [65]. The following introduces key aspects in the software's architectural design.

The Amsterdam Modeling Suite (AMS) collects different computational models of the PES into one package. A noteworthy feature of AMS is the separation between models – called *Engines* in AMS – and a so-called *Driver*. While the main function of Engines is to provide a PES (*cf.* Eq. 2.2), the Driver determines which specific points on it to sample. Sampling, in turn, depends on the task requested by the user. For example, a geometry optimization ideally results in the exploration of geometries with consecutively lower energies. Ultimately, the Driver is responsible for sampling the PES such that relevant physicochemical properties can be computed at relevant configurations of the chemical system. This setup allows users to treat Engines in a modular fashion: Given the same application, no additional input aside from the Engine needs to be changed.

ParAMS inherits this modularity, conceptually making the package applicable to all Engines supported by AMS. This results in the ability to mix and match different Engines for the purposes of reference data calculation and parameter fitting. Keeping the training set separate from the Engine also allows for functionality outside of parameter fitting, for example when comparing the quality of multiple models for a given application.

The fundamental ParAMS workflow is illustrated in Figure 5.1, showing how the loss function value $\mathcal{L}(y, \hat{y})$ is calculated starting from a chemical structure. Beginning at the top of the figure, the user defines a job as a combination of a system's geometry and the desired settings that will be passed to the AMS Driver. The settings include a computational task such as geometry optimization, PES scan or molecular dynamics. In the next step, the job can be combined with any Engine that is available within AMS. The job-engine pair is a complete instruction that is passed to AMS, which executes the calculation and returns a results object. Finally, as per the user's definition, ParAMS extracts the relevant properties from the calculated results and presents them as an input to a loss function. Examples for such properties could be system

5

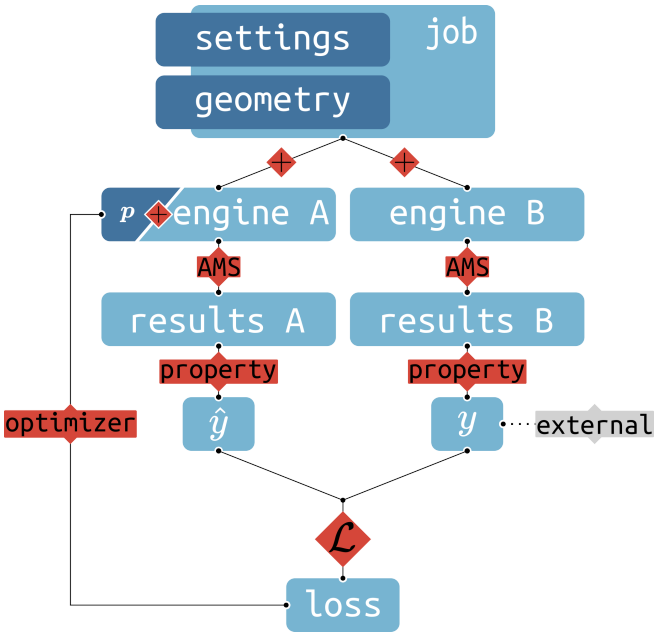


FIGURE 5.1: Flowchart illustrating how, starting from an input geometry and settings (top), ParAMS computes the loss function value (bottom). Parameter optimization implies iterative evaluation of the left-hand side loop. Operators depicted in red, inputs and outputs in blue.

energies, nuclear gradients or updated geometries. The complete process can be summarized as calculating a property P of job J with an Engine E. Calculation of a loss requires a second value: The same property computed with a different Engine. If interested in the comparison of two computational methods, all we have to do is replace the Engine and follow the same workflow once again, as illustrated by the two columns in Figure 5.1. Alternatively, values can also be provided directly to ParAMS. In this case, no computations are performed and the workflow is shortened to the last step, as indicated by the *external* operator in the figure. This is relevant when comparing to results from experiments or previous calculations.

Figure 5.1 also illustrates at which part in the workflow the parameter vector $\boldsymbol{p}$ comes into play. Here, Engine A is empirical as it requires a set of parameters. It is easy to see how optimization algorithms interface with the rest of the package: After calculating the loss, new parameters can be suggested. The loop on the left-hand side is iteratively repeated until an optimal solution is found. Note that the figure describes the flow for a single reference and predicted value. Expansion to multiple entries in the training set happens analogously, by extracting an arbitrary number of properties from an arbitrary number of computed jobs.

The following section contains our first publication, introducing the ParAMS package. For a complete documentation of the package along with examples and tutorials, please refer to Appendix B.

ParAMS: Parameter Optimization for Atomistic and Molecular Simulations

L. Komissarov, R. Rüger, M. Hellström and T. Verstraelen

J. Chem. Inf. Model. **2021**, *61*, 8, 3737–3743
DOI: 10.1021/acs.jcim.1c00333

L.K. and R.R. developed the ParAMS package. L.K.and M.H. performed the case studies. L.K., M.H., and T.V. wrote the paper. T.V. oversaw the project.

JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Application Note

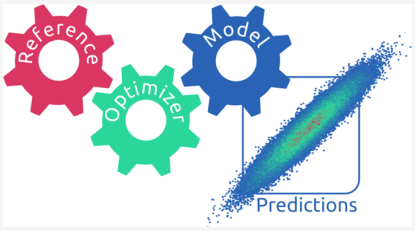
ParAMS: Parameter Optimization for Atomistic and Molecular Simulations

Leonid Komissarov, Robert Rüger, Matti Hellström, and Toon Verstraelen*

Cite This: https://doi.org/10.1021/acs.jcim.1c00333 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: This work introduces ParAMS—a versatile Python package that aims to make parametrization workflows in computational chemistry and physics more accessible, transparent, and reproducible. We demonstrate how ParAMS facilitates the parameter optimization for potential energy surface (PES) models, which can otherwise be a tedious specialist task. Because of the package's modular structure, various functionality can be easily combined to implement a diversity of parameter optimization protocols. For example, the choice of PES model and the parameter optimization algorithm can be selected independently. An illustration of ParAMS' strengths is provided in two case studies: (i) a density functional-based tight binding (DFTB) repulsive potential for the inorganic ionic crystal ZnO and (ii) a ReaxFF force field for the simulation of organic disulfides.



INTRODUCTION

Throughout the years, the use of predictive computational models has become standard practice in many professional fields such as logistics, economics, and R&D,[1−5] with computational chemistry being no exception. Predictive models in this field often approximate the potential energy surface (PES) and derived properties for a given chemical system and can be broadly categorized based on their level of theory; quantum mechanical (QM) approaches such as wave function or density functional theory explicitly model the electronic structure, generally resulting in a high accuracy and broad applicability. However, explicit electronic treatments come with a high computational price tag, making calculations unfeasible for larger systems. This limitation can be overcome by introducing more empiricism to the model on the electronic (e.g., tight binding), atomic (e.g., molecular mechanics), or molecular (e.g., coarse-graining) level. Such empirical models attempt to strike a balance between speed and prediction accuracy by approximating the PES description through the introduction of parameters. Prominent examples include approximate functionals in density functional theory (DFT),[6−9] density functional tight binding (DFTB),[10−14] machine learning (ML) potentials,[15−18] and force fields (FF).[19−22]

While some of the best empirical models can closely approximate higher-level QM theories at only a fraction of the computational time,[23,24] their quality strongly varies with different sets of parameters.[25−29] Additionally, many empirical models can only deliver accurate results for a comparably limited chemical space, either due to their functional form or parameters being specific to certain (combinations of) elements. These limitations can lead to the existence of multiple parameter sets based on the desired application; for example, ReaxFF has distinct families of combustion or condensed phase parametrizations.[30] Such lack of general parameters gives rise to the research field of parameter fitting,[25,31−34] where the task is to find an optimal parameter set, given training data constructed from chemical systems and their properties of interest. Although the fitting process can be an appealing solution to the above shortcomings, its practical implementation remains hardly accessible to the broader audience and instead is almost exclusively carried out by specialized research groups. In our experience, the majority of researchers, although being interested in individual parameter fitting, are discouraged by the high barrier that comes with it. The main reason for this being a lack of generalization and transparency: (1) Training data often come in a variety of formats. (2) Optimizers expect a different input all together. (3) The format in which parameters are stored is specific to each method.[35−38] The combination of these oftentimes results in works that can be hardly comprehended and reproduced by third parties. In an effort to address the above issues, we introduce the ParAMS scripting package for Python. The following section briefly summarizes the architecture of ParAMS, and we refer to the documentation for further technical details.[39] The Results section demonstrates how the package can be used to (i) generate density functional-based tight binding (DFTB) two-body repulsive

Received: March 25, 2021

   https://doi.org/10.1021/acs.jcim.1c00333
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

Journal of Chemical Information and Modeling pubs.acs.org/jcim Application Note

potentials for an ionic material and (ii) reparametrize a ReaxFF reactive force field[21,22] for organic disulfides. Step-by-step Jupyter notebooks for the two case studies are provided as Supporting Information, as well as on GitHub.[40] Additional application examples are available in the package's documentation.[39] The final section concludes with a summary and an outlook on future work.

IMPLEMENTATION

ParAMS follows a modular package structure with well-defined application programming interfaces (APIs), which allows components to be treated independently. This is essential for future development, as individual submodules can be easily worked on and extended. We describe the main components and their functionality below. For a mathematical description of the functionality as well as an additional explanation of the syntax, please refer to section S1 of the Supporting Information.

Job Collection and Data Sets. Job collection and data sets classes are responsible for the input/output (IO) of relevant data. In the context of ParAMS, these are collections of chemical systems, properties, and settings alongside optional metadata. A job collection clearly defines job entries by combining systems and settings. The data set defines which properties of a job are relevant to the optimization and stores the reference values of all entries in a vector $\boldsymbol{y}$. Reference values can be added from any source: experimental/external results or a high-level calculation. To ensure reproducibility and ease of use, ParAMS makes use of the YAML data serialization format[41] as the default for all IO operations.

Extractors. Extractors tell ParAMS how to extract a property of interest P from a calculated job. Technically, they are small standalone Python modules that read the native (e.g., stream, text or binary file) output of the Amsterdam Modeling Suite[42,43] (AMS) into Python variables. Examples for implemented extractors are interatomic distances, valence angles, dihedral angles, atomic charges, reaction energies, linear transit energy profiles, lattice vectors, bulk moduli, atomic forces, stress tensors, Hessian matrices, and vibrational frequencies. New extractors are easily written by the user, effectively allowing any property that can be calculated with AMS to be fitted within the scope of ParAMS. Extractors also support more elaborate cases that require additional processing before a comparison. This is, for example, needed when computing the minimal root-mean-square deviation of atomic positions.

Loss Functions. Loss functions implement various metrics that describe the distance between two vectors. In the context of parameter fitting, a vector consisting of all reference values $\boldsymbol{y}$ has to be compared to the predictions vector $\hat{\boldsymbol{y}}$, as generated given a specific set of parameters, in order to measure the quality of the fit. An additional, user-defined weights vector is passed to the loss function to fine-tune the importance of each entry in the data set (see the Supporting Information for a mathematical definition of all relevant components). Implemented metrics are least absolute error (LAE), mean absolute error (MAE), root-mean-square error (RMSE), and residual sum of squares (RSS). Additionally, user-defined metrics are supported.

Optimizers. Optimizers provide a unified interface to a variety of optimization algorithms. Currently, the following are supported: Covariance Matrix Adaptation Evolution Strategy (CMA-ES),[44−46] Adaptive Rate Monte Carlo (ARMC),[47] and optimizers available through the Nevergrad[48] and SciPy[49] packages. To guarantee parametrization support for a wider range of models, the current version of ParAMS is designed to work with gradient-free optimization algorithms only.

Parameter Interfaces. Parameter interfaces translate the parameter vector $\boldsymbol{x}$ into the native format of the empirical model (e.g., a file on disk). Any existing parameter interface can be parametrized. At the time of writing, ParAMS supports interfaces to ReaxFF,[50] SCC-DFTB repulsive potentials, GFN1-xTB,[13] and Lennard-Jones potentials.

Callbacks. Callbacks allow the interaction with a parametrization at runtime. Such interactions can be progress loggers, timeouts, early stopping criteria, or plotting functions.[51]

A command line interface is provided for users who do not wish to spend much time writing their own parametrization scripts. It allows the setup and execution of the most common tasks through a configuration file.

Figure 1 shows the general parametrization loop and highlights the main input−output relationships.

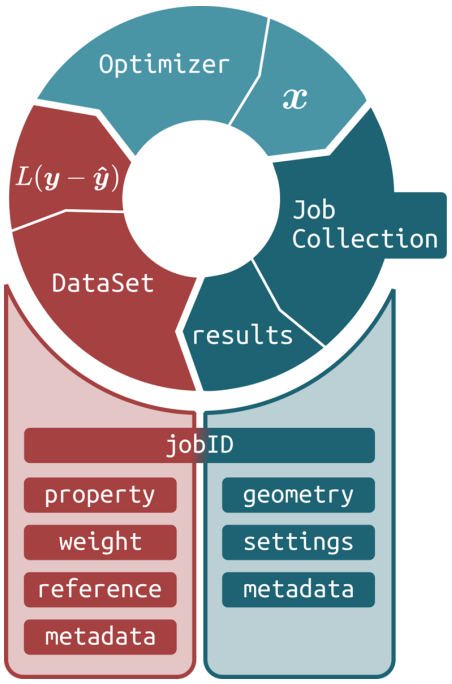


Figure 1. Schematic representation of the ParAMS parametrization loop. Highlighted is the interplay between the three main components: Optimizer, Job Collection, and Data Set. Every parameter set suggested by the optimizer produces different job results, which are evaluated by the Data Set based on a unique jobID. The results are then compared with the reference values, weighted, and combined into the loss function value. A more detailed description is available in the package documentation.

The implementation allows users already experienced in other data science packages to prepare inputs and process results in a familiar way. Techniques like training set and validation set splitting, cross-validation, early stopping, batching, or outlier detection are supported out of the box. Parameter constraints can be further used to limit the search

https://doi.org/10.1021/acs.jcim.1c00333
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

space of a problem. In addition to regular box constraints, users can express inequality constraints involving multiple parameters (e.g., $x_1 \leq x_2$), as is demonstrated in the ReaxFF example application. Moreover, since the signatures of all classes in a submodule are the same, different models and optimizers can be effortlessly compared and deployed (e.g., comparison of different levels of theory in Table 1). Daily regression tests are performed to guarantee an error-free functionality of the package. ParAMS implements two levels of parallelism. Multiple parameter vectors and multiple jobs per vector can be evaluated at the same time, resulting in workloads that can be distributed effectively. By default, ParAMS will prioritize parallelization over parameter vectors (running all jobs serially per vector). Optimization algorithms that are able to suggest and process candidates asynchronously will benefit from this setup. In the case of population-based optimizers, possible idle times can still occur when the evaluation of some candidate vectors is slower than others. In contrast, slow models that scale well with the number of CPU cores can be set up to run in parallel, while all parameter vectors are evaluated sequentially. The optimal settings are often a function of multiple factors, such as the chosen computational model, optimization algorithm, or number and type of jobs. Consequently, ParAMS allows the user to fine-tune the parallelization options for optimal performance. This approach is also available when running on high-performance computing clusters. Effective hardware utilization is currently limited to one node per optimization instance—an issue that will be fixed in the near future.

RESULTS

DFTB Two-Body Repulsive Potential Parametrization.

Here, we illustrate how the ParAMS package can be used to train a two-body SCC-DFTB repulsive potential.[10] For simplicity, we use ZnO, for which several previous parametrizations already exist in the literature, for example, the *znorg-0-1*[52] and *znopt*[53] parameter sets. We approximately follow the approach from ref 53, (znopt), in which the authors reused the electronic parameters from znorg-0-1[52] and reparametrized the two-body Zn−O repulsive potential to reference data calculated for the wurtzite and rocksalt polymorphs of ZnO. With the znorg-0-1 parameters, the rocksalt polymorph is predicted to be more stable than the wurtzite polymorph, but in experiments and DFT calculations, the opposite is true, which motivates the reparametrization.

The entire code needed for the parametrization is provided in the Supporting Information.[40] It fits the Zn−O pairwise repulsive potential V^{rep} as a tapered double exponential function of the form

$$V^{\text{rep}}(r) = [A_0 \exp(-A_1 r) + A_2 \exp(-A_3 r)] f^{\text{cut}}(r) \quad (1)$$

where A_0, A_1, A_2, and A_3 are the parameters, and $f^{\text{cut}}(r)$ is a tapering function of the form

$$f^{\text{cut}}(r) = \frac{1}{2}\left(\cos\left(\frac{\pi r}{r_{\text{cut}}}\right) + 1\right) \quad (2)$$

with the cutoff distance r_{cut} = 5.67 bohr.

With ParAMS, it is possible to either directly define the reference values for the training set (for example, from literature values) or to automatically calculate them if no reference values have been given. Here, we illustrate the second approach and perform the reference calculations using the periodic DFT code BAND[54] in the Amsterdam Modeling Suite[42] (AMS).

The training set comprises the a and c lattice parameters of wurtzite ZnO and the bulk modulus B_0 of wurtzite ZnO, as well as the relative energies of the wurtzite and rocksalt polymorphs of ZnO, $\Delta E = E_{\text{wurtzite}} - E_{\text{rocksalt}}$ (per ZnO formula unit). We do not need to specify the reference values themselves, as they are automatically calculated by ParAMS.

The job collection contains two jobs: lattice optimizations of the wurtzite and rocksalt polymorphs. From these two jobs, the a, c, B_0, and ΔE quantities can be extracted. For wurtzite, B_0 can be extracted by requesting that the elastic tensor be calculated at the end of the lattice optimization.

The reference DFT calculations were run with the PBE exchange-correlation functional, a triple-ζ (TZP) basis set, and "Good" numerical quality (dense k-space and integration grids). For the parametrized DFTB engine, a "Good" (dense) k-space grid was also used, since the results of lattice optimizations can be quite sensitive to the k-space grid.

The optimization was done with the Nelder−Mead algorithm[55] from scipy, with a sum-of-squared-errors loss function. The smallest loss function value was obtained for A_0 = 0.45, A_1 = 1.01, A_2 = 0.25, and A_3 = 0.40. The resulting repulsive potential is shown in Figure 2. Typical Zn−O

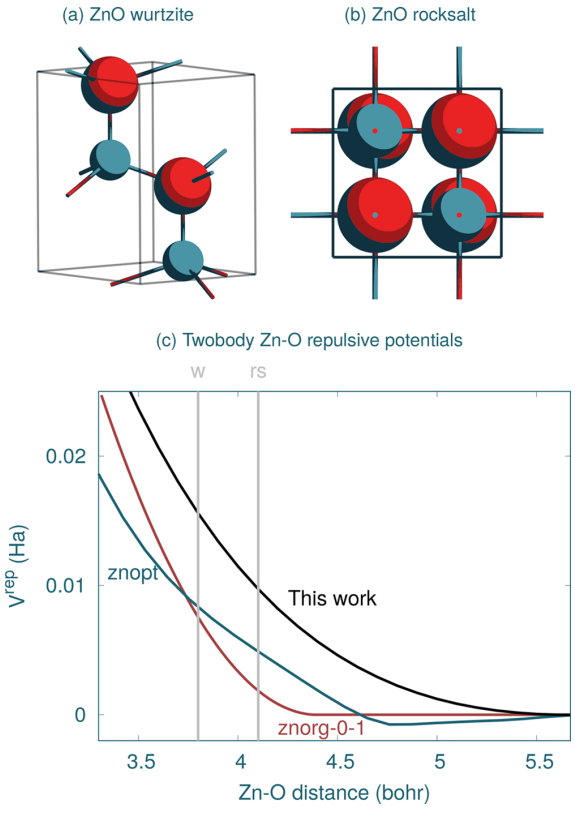


Figure 2. ZnO parametrization data. (a, b) Pictures of the unit cells of ZnO wurtzite and rocksalt polymorphs. Zn is shown in blue and O in red. (c) DFTB Zn−O pairwise repulsive potentials for the znorg-0-1[52] (red), znopt[53] (blue), and refitted repulsive potential from this work (black). The gray lines mark typical Zn−O distances in the wurtzite (w) and rocksalt (rs) polymorphs, respectively.

https://doi.org/10.1021/acs.jcim.1c00333
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

distances in wurtzite ("w") and rocksalt ("rs") are indicated with gray lines at 3.8 and 4.1 bohr, respectively. With znorg-0-1 (red line), the repulsive potential decays very rapidly between the typical wurtzite and rocksalt distances. This decrease is not as pronounced with either the potential in this work (black line) or znopt (blue line). This affects the relative stability of the wurtzite and rocksalt ZnO polymorphs.

Table 1 compares the resulting ZnO properties from the parametrization in this work to the DFT reference data, as well as previous DFTB ZnO parametrizations (note: znorg-0-1 was not parametrized to the DFT data in Table 1, and znopt was parametrized to also reproduce some adsorption energies). The repulsive potential in this work closely reproduces the wurtzite lattice parameters a and c and the relative energy ΔE and provides a good estimate of the bulk modulus, compared to the DFT reference to which it was trained.

Table 1. Calculated ZnO Wurtzite Lattice Constants a and c, Bulk Modulus B_0, and Energy Relative to the Rocksalt Polymorph $\Delta E = E_{\text{wurtzite}} - E_{\text{rocksalt}}$ (per ZnO Formula Unit)

	a (Å)	c (Å)	B_0 (GPa)	ΔE (eV)
DFT (this work)	3.30	5.32	126	−0.24
DFTB (this work)	3.28	5.34	146	−0.24
DFT[53]	3.29	5.31	129	−0.30
DFTB, znopt[53]	3.21	5.25	161	−0.32
DFTB, znorg-0-1[52]	3.29	5.38	161	+0.14
UFF	2.90	4.73	199	−16.2

With ParAMS, it is additionally possible to evaluate the loss function, and individual training set entries, using any of the engines in the Amsterdam Modeling Suite. For comparison, Table 1 also gives the corresponding quantities for the UFF (Universal Force Field) engine, which performs significantly worse than the DFTB parametrizations for these quantities.

ReaxFF Parametrization. ReaxFF is another contemporary example of an empirical model.[21,22] This formalism has been applied to a wide range of chemical problems and consequently has seen a lot of new parameter development[56−58] (for a general overview of ReaxFF and its development, see Senftle et al.[30]). In this section, we demonstrate the parametrization of ReaxFF with a training set previously published by Müller and Hartke (MH).[58] Unlike in the DFTB example, we are using MH's previously published CASPT2 data, thus saving the need to perform expensive reference calculations. The optimized parameter vector $\boldsymbol{x}^*$, as found by MH, is called Mue2016. An overview of the training set can be found in Table S1 of the Supporting Information. It features a total of 231 geometries needed for the computation of 4875 chemical properties. Additionally, MH included a validation set to check for overfitting. Examples of three structures included in the data are depicted in Figure S1 of the Supporting Information, showing cyclopentathione, diphenyl disulfide, and dimethyl disulfide. Prior to the parameter optimization, we evaluate the training and validation sets with the Mue2016 ReaxFF parameters and report sum-of-squared-error (SSE) losses of 14,441 and 14,451, respectively. Note that in the original publication, MH report a training set loss of 12,400,[58] while in a more recent work, Shchygol et al. calculate a loss of 16,300.[25] Such differences are expected, because discontinuities in gradients and energies inherent to ReaxFF and software improvements (mostly related to geometry optimization) may result in different optimized geometries.[25,59]

In our setup, we use the covariance matrix adaptation evolution strategy (CMA-ES)[44−46] as the optimization algorithm with Mue2016 as the initial point. CMA-ES is gradient free and relies on a population to sample new parameter vectors from an adapted, n-dimensional Gaussian. It does not require additional hyperparameters other than a population size and an initial width of the Gaussian distribution, σ. Here, we use a population size of 36 and an initial σ of 0.3. Furthermore, we limit the optimization to 24 h and set up an early stopping mechanism based on the validation set. The optimization is set up to stop early only if there has been no improvement in the validation set error for the last 6000 evaluations.

Rather than optimizing the same 87 parameters as MH, we perform a one-dimensional scan on all parameters and select the 35 most sensitive with respect to the training set; although Mue2016 is a set of 701 parameters in total, only a subset of these significantly affects the overall cost function value. This is, for example, the case when a model includes parameters for each chemical element (e.g., C, H, O), but the total training set of systems R can be constructed from fewer elements (e.g., C, H). In such cases, the dimensionality of the problem can be reduced by scanning for a relevant parameter subset which yields the biggest change in the cost function value. The simplest setting, which we used in this case study, only modifies one parameter at a time to determine its influence on the objective function. It is also possible to scan all parameter combinations to discover coupling between parameters, albeit at a highly increased computational cost. Here, the explicit selection of this parameter subset is presented as an example of ParAMS' features. No other subset sizes have been tested. Out of the 35 parameters selected this way, 16 have also been optimized by MH. We list all optimized parameters in the provided Python notebooks.[40]

Parameter bounds are set to be relative to the initial values such that $\boldsymbol{x}_\pm = \boldsymbol{x}_0 \pm 0.2|\boldsymbol{x}_0|$. In addition to box constraints, ParAMS enables a definition of inequality constraints. As the ReaxFF formalism works with bond orders, we limit the parameters responsible for the covalent radii of σ, π, and $\pi\pi$ bonds to $r_0^\sigma \geq r_0^\pi \geq r_0^{\pi\pi}$ for every atom and atom pair defined in the force field. This approach effectively limits the search space and is available in combination with all optimizers.

A summary of all settings is provided in Table S2 in the Supporting Information. To compensate for the randomness of CMA-ES, we repeat the optimization setup nine times. For the best solution, we report improved training set and validation set losses of 11,877 and 5377, respectively. We make this work's optimized parameter set available through the Supporting Information under the title MueParAMS. Correlation plots between reference and predicted values for the new parameters are presented in Figure 3, showing very good agreement to the reference data. Moreover, Figure 4 compares the S−S dissociation curve of diphenyl disulfide, as computed with Mue2016 and the new MueParAMS parameters, showing an improved agreement to the reference data for this case.

■ SUMMARY AND OUTLOOK

With ParAMS, we have presented a modern Python package, supporting versatile parametrization workflows with minimal effort. Its integration with the Amsterdam Modeling Suite adds a high amount of flexibility through the number of properties

https://doi.org/10.1021/acs.jcim.1c00333
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

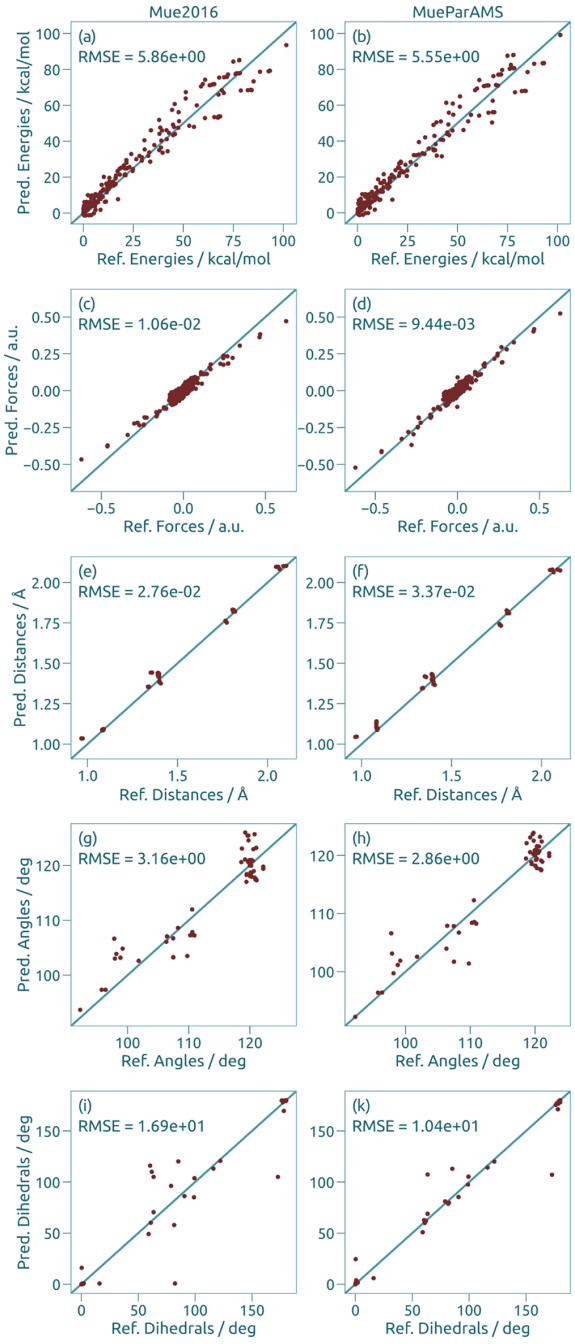


Figure 3. Training data correlation plots. (a, b) Energy differences, (c, d) atomic forces, (e, f) atomic distances, (g, h) interatomic angles, and (i, k) dihedral angles as calculated with Mue2016 (left) and MueParAMS (right) parameters. Reaction energies and internal coordinates are compared after geometry optimization. *X* and *Y* axes depict reference and predicted values, respectively.

that can be fitted alongside the support for multiple codes when it comes to the model, optimization algorithm, and reference data selection. Features such as highly customizable optimizations, support for multiple validation sets, or intuitive

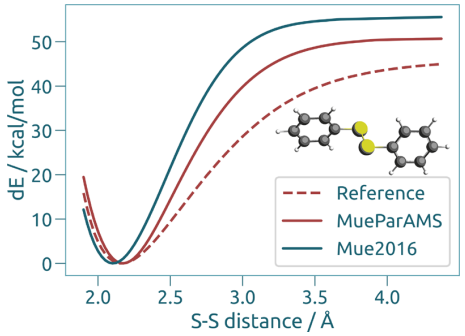


Figure 4. S−S dissociation curve of diphenyl disulfide. Computed with Mue2016 and MueParAMS ReaxFF parametrizations. For details about the reference data, see ref 58.

processing of data aim to make ParAMS accessible to both advanced and less experienced users. At the same time, developers can easily extend existing functionality. We showed how an SCC-DFTB repulsive potential could easily be parametrized for the inorganic crystal ZnO. The reference data were calculated automatically using a DFT engine within AMS. This example application also demonstrates how ParAMS can be used to compare the accuracy of different chemical simulation packages given the same training set. We also demonstrated how the package can be used to easily process, set up, and start a fitting procedure for ReaxFF. Using previously published data by Müller and Hartke,[58] we were able to find parameters that produce a considerably lower error for the validation set while maintaining a similar accuracy in the training data. In the future, we hope to extend the number of empirical models that can be fitted with ParAMS and further improve the ease of use through the introduction of additional shortcut functions for training set building. We also expect additions in other functionality such as optimization algorithms or extractors based on user feedback and wishes as the project matures. The package is included in all AMS releases since 2020.

DATA AND SOFTWARE AVAILABILITY

All data needed to reproduce the examples are available at https://www.doi.org/10.5281/zenodo.4629706. The package's documentation is available at https://www.scm.com/doc.trunk/params. ParAMS is distributed with the Amsterdam Modeling Suite, for which a free trial can be requested at https://www.scm.com.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.1c00333.

Mathematical description of the optimization problem, summary of reference data published by Müller and Hartke (ref 1) used in the ReaxFF example, settings used for the ReaxFF parametrization, and visualization of some structures in the Müller and Hartke training set (PDF)

https://doi.org/10.1021/acs.jcim.1c00333
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

Journal of Chemical Information and Modeling pubs.acs.org/jcim Application Note

AUTHOR INFORMATION

Corresponding Author

Toon Verstraelen − *Center for Molecular Modeling (CMM), Ghent University, B-9052 Ghent, Belgium;* orcid.org/0000-0001-9288-5608; Email: toon.verstraelen@ugent.be

Authors

Leonid Komissarov − *Center for Molecular Modeling (CMM), Ghent University, B-9052 Ghent, Belgium; Software for Chemistry & Materials (SCM) B.V., 1081 HV Amsterdam, The Netherlands;* orcid.org/0000-0001-6011-1632

Robert Rüger − *Software for Chemistry & Materials (SCM) B.V., 1081 HV Amsterdam, The Netherlands*

Matti Hellström − *Software for Chemistry & Materials (SCM) B.V., 1081 HV Amsterdam, The Netherlands;* orcid.org/0000-0003-3053-5658

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.1c00333

Author Contributions

L.K. and R.R. developed the ParAMS Python package. L.K. and M.H. performed the case studies. L.K., M.H., and T.V. wrote the paper. T.V. oversaw the project. All authors read and approved the final manuscript.

Notes

The authors declare the following competing financial interest(s): Authors L.K., R.R. and M.H. were employed by the company Software for Chemistry and Materials (SCM). SCM develops and commercializes the Amsterdam Modeling Suite, of which ParAMS is a new module.

ACKNOWLEDGMENTS

We thank Michal Handzlik for the initial design of the ParAMS package and Dr. Tomáš Trnka for the implementation of the AMSWorker interface, resulting in a considerable computational speed-up. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 814143 (L.K. and T.V.) and No. 798129 (M.H). T.V. also acknowledges funding of the research board of Ghent University. R.R. and M.H. have received funding from The Netherlands Enterprise Agency (RVO) and Stimulus under the MIT R&D Collaboration programme, Project No. PROJ-02612. The computational resources (Stevin Supercomputer Infrastructure) and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by Ghent University, FWO, and the Flemish Government−Department EWI.

REFERENCES

(1) Bruzda, J. Multistep Quantile Forecasts for Supply Chain and Logistics Operations: bootstrapping, the GARCH Model and Quantile Regression Based Approaches. *Cent. Eur. J. Oper. Res.* **2020**, *28*, 309−336.

(2) Wolfers, J.; Zitzewitz, E. Prediction Markets. *J. Econ. Perspect.* **2004**, *18*, 107−126.

(3) Lewis, B. P.; Shih, I-h.; Jones-Rhoades, M. W.; Bartel, D. P.; Burge, C. B. Prediction of Mammalian MicroRNA Targets. *Cell* **2003**, *115*, 787−798.

(4) Wilson, R. L.; Sharda, R. Bankruptcy Prediction Using Neural Networks. *Decis. Support Syst.* **1994**, *11*, 545−557.

(5) Geller, R. J. Earthquake Prediction: A Critical Review. *Geophys. J. Int.* **1997**, *131*, 425−450.

(6) Becke, A. D. A. New Mixing of Hartree−Fock and Local Density-functional Theories. *J. Chem. Phys.* **1993**, *98*, 1372−1377.

(7) Lee, C.; Yang, W.; Parr, R. G. Development of the Colle-Salvetti Correlation-energy Formula Into a Functional of the Electron Density. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1988**, *37*, 785−789.

(8) Stephens, P. J.; Devlin, F. J.; Chabalowski, C. F.; Frisch, M. J. Ab Initio Calculation of Vibrational Absorption and Circular Dichroism Spectra Using Density Functional Force Fields. *J. Phys. Chem.* **1994**, *98*, 11623−11627.

(9) Chai, J.-D.; Head-Gordon, M. Long-range Corrected Hybrid Density Functionals with Damped Atom−atom Dispersion Corrections. *Phys. Chem. Chem. Phys.* **2008**, *10*, 6615.

(10) Elstner, M.; Porezag, D.; Jungnickel, G.; Elsner, J.; Haugk, M.; Frauenheim, T.; Suhai, S.; Seifert, G. Self-Consistent-Charge Density-functional Tight-binding Method for Simulations of Complex Materials Properties. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1998**, *58*, 7260−7268.

(11) Yang; Yu, H.; York, D.; Cui, Q.; Elstner, M. Extension of the Self-Consistent-Charge Density-Functional Tight-Binding Method: Third-Order Expansion of the Density Functional Theory Total Energy and Introduction of a Modified Effective Coulomb Interaction. *J. Phys. Chem. A* **2007**, *111*, 10861−10873.

(12) Gaus, M.; Cui, Q.; Elstner, M. DFTB3: Extension of the Self-Consistent-Charge Density-Functional Tight-Binding Method (SCC-DFTB). *J. Chem. Theory Comput.* **2011**, *7*, 931−948.

(13) 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.* **2017**, *13*, 1989−2009.

(14) Bannwarth, C.; Ehlert, S.; Grimme, S. GFN2-xTB—An Accurate and Broadly Parametrized Self-Consistent Tight-Binding Quantum Chemical Method With Multipole Electrostatics and Density-Dependent Dispersion Contributions. *J. Chem. Theory Comput.* **2019**, *15*, 1652−1671.

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

(16) Schütt, K. T.; Sauceda, H. E.; Kindermans, P.-J.; Tkatchenko, A.; Müller, K.-R. SchNet − A Deep Learning Architecture for Molecules and Materials. *J. Chem. Phys.* **2018**, *148*, 241722.

(17) Smith, J. S.; Nebgen, B.; Lubbers, N.; Isayev, O.; Roitberg, A. E. Less is More: Sampling Chemical Space with Active Learning. *J. Chem. Phys.* **2018**, *148*, 241733.

(18) Shao, Y.; Hellström, M.; Mitev, P. D.; Knijff, L.; Zhang, C. PiNN: A Python Library for Building Atomic Neural Networks of Molecules and Materials. *J. Chem. Inf. Model.* **2020**, *60*, 1184−1193.

(19) Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A. A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules. *J. Am. Chem. Soc.* **1995**, *117*, 5179−5197.

(20) MacKerell, A. D.; Bashford, D.; Bellott, M.; Dunbrack, R. L.; Evanseck, J. D.; Field, M. J.; Fischer, S.; Gao, J.; Guo, H.; Ha, S.; Joseph-McCarthy, D.; Kuchnir, L.; Kuczera, K.; Lau, F. T. K.; Mattos, C.; Michnick, S.; Ngo, T.; Nguyen, D. T.; Prodhom, B.; Reiher, W. E.; Roux, B.; Schlenkrich, M.; Smith, J. C.; Stote, R.; Straub, J.; Watanabe, M.; Wiórkiewicz-Kuczera, J.; Yin, D.; Karplus, M. All-Atom Empirical Potential for Molecular Modeling and Dynamics Studies of Proteins. *J. Phys. Chem. B* **1998**, *102*, 3586−3616.

(21) van Duin, A. C. T.; Dasgupta, S.; Lorant, F.; Goddard, W. A. ReaxFF: A Reactive Force Field for Hydrocarbons. *J. Phys. Chem. A* **2001**, *105*, 9396−9409.

(22) Chenoweth, K.; van Duin, A. C. T.; Goddard, W. A. ReaxFF Reactive Force Field for Molecular Dynamics Simulations of Hydrocarbon Oxidation. *J. Phys. Chem. A* **2008**, *112*, 1040−1053.

(23) Smith, J. S.; Nebgen, B. T.; Zubatyuk, R.; Lubbers, N.; Devereux, C.; Barros, K.; Tretiak, S.; Isayev, O.; Roitberg, A.

Approaching Coupled Cluster Accuracy With a General-Purpose Neural Network Potential Through Transfer Learning. ChemRxiv, 2019. https://chemrxiv.org/articles/Outsmarting_Quantum_Chemistry_Through_Transfer_Learning/6744440 (accessed May 2021).
(24) Bannwarth, C.; Ehlert, S.; Grimme, S. GFN2-xTB—An Accurate and Broadly Parametrized Self-Consistent Tight-Binding Quantum Chemical Method With Multipole Electrostatics and Density-Dependent Dispersion Contributions. *J. Chem. Theory Comput.* **2019**, *15*, 1652−1671.
(25) Shchygol, G.; Yakovlev, A.; Trnka, T.; van Duin, A. C. T.; Verstraelen, T. ReaxFF Parameter Optimization with Monte-Carlo and Evolutionary Algorithms: Guidelines and Insights. *J. Chem. Theory Comput.* **2019**, *15*, 6799−6812.
(26) Dieterich, J. M.; Hartke, B. OGOLEM: Global Cluster Structure Optimisation for Arbitrary Mixtures of Flexible Molecules. A Multiscaling, Object-oriented Approach. *Mol. Phys.* **2010**, *108*, 279−291.
(27) Wang, L.-P.; Chen, J.; Van Voorhis, T. Systematic Parametrization of Polarizable Force Fields From Quantum Chemistry Data. *J. Chem. Theory Comput.* **2013**, *9*, 452−460.
(28) Brommer, P.; Kiselev, A.; Schopf, D.; Beck, P.; Roth, J.; Trebin, H.-R. Classical Interaction Potentials for Diverse Materials From Ab Initio Data: a Review Ofpotfit. *Modell. Simul. Mater. Sci. Eng.* **2015**, *23*, 074002.
(29) Jaramillo-Botero, A.; Naserifar, S.; Goddard, W. A. General Multiobjective Force Field Optimization Framework, With Application to Reactive Force Fields for Silicon Carbide. *J. Chem. Theory Comput.* **2014**, *10*, 1426−1439.
(30) Senftle, T. P.; Hong, S.; Islam, M. M.; Kylasa, S. B.; Zheng, Y.; Shin, Y. K.; Junkermeier, C.; Engel-Herbert, R.; Janik, M. J.; Aktulga, H. M.; Verstraelen, T.; Grama, A.; van Duin, A. C. T. a. The ReaxFF Reactive Force-field: Development, Applications and Future Directions. *npj Comput. Mater.* **2016**, *2*, 15011.
(31) Guvench, O.; Hatcher, E.; Venable, R. M.; Pastor, R. W.; MacKerell, A. D. CHARMM Additive All-Atom Force Field for Glycosidic Linkages Between Hexopyranoses. *J. Chem. Theory Comput.* **2009**, *5*, 2353−2370.
(32) Gaus, M.; Chou, C.-P.; Witek, H.; Elstner, M. Automatized Parametrization of SCC-DFTB Repulsive Potentials: Application to Hydrocarbons. *J. Phys. Chem. A* **2009**, *113*, 11866−11881.
(33) van Beest, B. W. H.; Kramer, G. J.; van Santen, R. A. Force Fields for Silicas and Aluminophosphates Based on ab Initio Calculations. *Phys. Rev. Lett.* **1990**, *64*, 1955−1958.
(34) Ashraf, C.; van Duin, A. C. Extension of the ReaxFF Combustion Force Field Toward Syngas Combustion and Initial Oxidation Kinetics. *J. Phys. Chem. A* **2017**, *121*, 1051−1068.
(35) van Duin, A. C. T. *ReaxFF User Manual*, 2002. https://www.scm.com/doc/ReaxFF/index.html (accessed May 2021).
(36) Wang, L.-P. *ForceBalance: Main Page*. http://leeping.github.io/forcebalance/doc/html/index.html (accessed May 2021).
(37) Dieterich, J. M.; Hartke, B. *OGOLEM.ORG*. https://www.ogolem.org/manual/ (accessed May 2021).
(38) Martinez, J. A.; Chernatynskiy, A.; Yilmaz, D. E.; Liang, T.; Sinnott, S. B.; Phillpot, S. R. Potential Optimization Software for Materials (POSMat). *Comput. Phys. Commun.* **2016**, *203*, 201−211.
(39) Komissarov, L.; Rüger, R. *ParAMS Documentation*. https://www.scm.com/doc.trunk/params/index.html (accessed May 2021).
(40) Komissarov, L.; Rüger, R.; Hellström, M.; Verstraelen, T. *ParAMS Supporting Information*, 2020. https://zenodo.org/record/4629706#.YJQdDqEpBaQ6 (accessed May 2021).
(41) döt Net, I.; Evans, C.; Ben-Kiki, O. *Official YAML Web Site*, 2019. https://yaml.org/ (accessed May 2021).
(42) Rüger, R.; Franchini, M.; Trnka, T.; Yakovlev, A. L.; van Lenthe, E.; Philipsen, P. H. T.; van Vuren, T.; Klumpers, B.; Soini, T. *Amsterdam Modeling Suite*, 2019. https://scm.com (accessed May 2021).
(43) van Duin, A. C. T.; Goddard, W. A.; Islam, M.; van Schoot, H.; Trnka, T.; Yakovlev, A. L. *ReaxFF 2019*, Vol. 4, 2019. https://www.scm.com/ (accessed May 2021).
(44) Hansen, N.; Ostermeier, A. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evol. Comput.* **2001**, *9*, 159−195.
(45) Hansen, N.; Kern, S. Evaluating the CMA Evolution Strategy on Multimodal Test Functions.In *Parallel Problem Solving from Nature PPSN VIII*; 8th International Conference, Birmingham, U.K., September 18−22, 2004; pp 282−291.
(46) Hansen, N.; Akimoto, Y.; Baudis, P. *CMA-ES/pycma on Github*, 2019. https://zenodo.org/record/2651072#.YJQg2KEpBaQ (accessed May 2021).
(47) Cosseddu, S.; Infante, I. Force Field Parametrization of Colloidal CdSe Nanocrystals Using an Adaptive Rate Monte Carlo Optimization Algorithm. *J. Chem. Theory Comput.* **2017**, *13*, 297−308.
(48) Rapin, J.; Teytaud, O. *Nevergrad - A Gradient-free Optimization Platform*, 2018.https://GitHub.com/FacebookResearch/Nevergrad (accessed May 2021).
(49) Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; van der Walt, S. J.; Brett, M.; Wilson, J.; Millman, K. J.; Mayorov, N.; Nelson, A. R. J.; Jones, E.; Kern, R.; Larson, E.; Carey, C J; Polat, I.; Feng, Y.; Moore, E. W.; VanderPlas, J.; Laxalde, D.; Perktold, J.; Cimrman, R.; Henriksen, I.; Quintero, E. A.; Harris, C. R.; Archibald, A. M.; Ribeiro, A. H.; Pedregosa, F.; van Mulbregt, P. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **2020**, *17*, 261−272.
(50) Kamat, A. M.; van Duin, A. C. T.; Yakovlev, A. Molecular Dynamics Simulations of Laser-Induced Incandescence of Soot Using an Extended ReaxFF Reactive Force Field. *J. Phys. Chem. A* **2010**, *114*, 12561−12572.
(51) Hunter, J. D. Matplotlib: A 2D Graphics Environment. *Comput. Sci. Eng.* **2007**, *9*, 90−95.
(52) Moreira, N. H.; Dolgonos, G.; Aradi, B.; da Rosa, A. L.; Frauenheim, T. Toward an Accurate Density-Functional Tight-Binding Description of Zinc-Containing Compounds. *J. Chem. Theory Comput.* **2009**, *5*, 605−614.
(53) Hellström, M.; Jorner, K.; Bryngelsson, M.; Huber, S. E.; Kullgren, J.; Frauenheim, T.; Broqvist, P. An SCC-DFTB Repulsive Potential for Various ZnO Polymorphs and the ZnO−Water System. *J. Phys. Chem. C* **2013**, *117*, 17004−17015.
(54) te Velde, G.; Baerends, E. J. Precise Density-functional Method for Periodic Structures. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1991**, *44*, 7888−7903.
(55) Nelder, J. A.; Mead, R. A Simplex Method for Function Minimization. *Comput. J.* **1965**, *7*, 308−313.
(56) Strachan, A.; van Duin, A. C. T.; Chakraborty, D.; Dasgupta, S.; Goddard, W. A. Shock Waves in High-Energy Materials: The Initial Chemical Events in Nitramine RDX. *Phys. Rev. Lett.* **2003**, *91*, 098301.
(57) Fogarty, J. C.; Aktulga, H. M.; Grama, A. Y.; van Duin, A. C. T.; Pandit, S. A. A Reactive Molecular Dynamics Simulation of the Silica-Water Interface. *J. Chem. Phys.* **2010**, *132*, 174704.
(58) Müller, J.; Hartke, B. ReaxFF Reactive Force Field for Disulfide Mechanochemistry, Fitted to Multireference ab Initio Data. *J. Chem. Theory Comput.* **2016**, *12*, 3913−3925.
(59) Furman, D.; Wales, D. J. Transforming the Accuracy and Numerical Stability of ReaxFF Reactive Force Fields. *J. Phys. Chem. Lett.* **2019**, *10*, 7215−7223.

https://doi.org/10.1021/acs.jcim.1c00333
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

6 Applications

Following the title of this thesis, this chapter presents two publications related to model optimization. Section 6.1 (Paper II) applies the ParAMS package to the reparametrization of the GFN1-xTB Hamiltonian [45]. In it, we show that the semi-empirical model fails to accurately describe small, isolated structures that contain silicon atoms. This is addressed by constructing a DFT reference data set with structures from the PubChem library [66, 67] and a consecutive fitting procedure that optimizes all GFN1-xTB parameters related to silicon, resulting in a more accurate parametrization.

In Section 6.2, Paper III describes a reference data set of bulk zeolite structures. Although previous chapters do not address the generation of reference data in detail, it is a crucial part of the fitting process: Researchers must be careful with its selection, as the quality of the reference data directly translates to the quality of the fitted model. Additionally, the generation of reference data is likely the most time-consuming step, considered that it originates from experiments or ab initio calculations (*cf.* Section 2.3). The above motivated us to publish a set of zeolite structures from the Database of Zeolite Structures [68], geometry-optimized on the DFT level.

Silicon being involved in both publications is no coincidence. Originally, our goal was to address the issue of collapsing periodic zeolite structures in the GFN1-xTB Hamiltonian. As displayed in Figure 6.1, computational tasks that

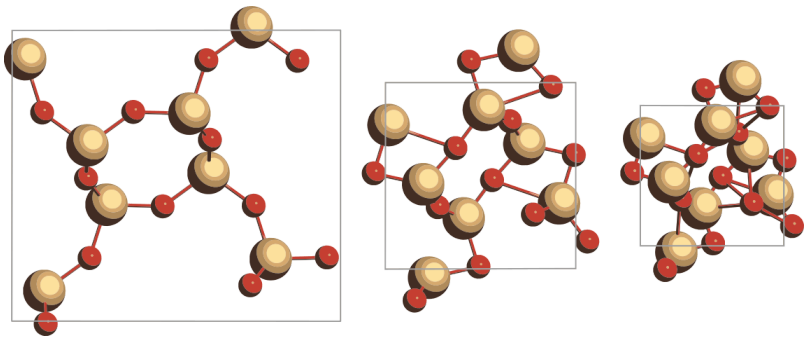


FIGURE 6.1: Three snapshots from the lattice optimization trajectory of the ABW framework, calculated with the original GFN1-xTB parameters. The model fails to describe periodic zeolite structures as they collapse. Oxygen atoms depicted in red, silicon in yellow. Grey boxes mark the unit cell edges.

allowed for the adjustment of cell parameters commonly resulted in unphysical geometries. Our initial approach was to fit the silicon parameters of GFN1-xTB to the reference data from Paper III (Section 6.2), and use the data from Paper II (Section 6.1) as a test set. However, such a workflow is subject to the initial parametrization performing reasonably well on the test set. As this was not the case, we decided to address the inaccurate predictions on the PubChem data in Paper II first, before moving on to the periodic zeolite structures from Paper III. Although a thorough investigation is still pending, preliminary results show that the optimized parameters from Paper II are able to address the issue of collapsing zeolites.

The issue of collapsing structures is not limited to zeolite structures. While we have observed the same behavior for inorganic salts, Raaijmakers *et al.* [69] also report it for perovskites. It is likely that other compound classes that need to be modeled under periodic boundary conditions are also affected.

Improving the Silicon Interactions of GFN-xTB

L. Komissarov and T. Verstraelen

J. Chem. Inf. Model. **2021**, *61*, 12, 5931-5937
DOI: 10.1021/acs.jcim.1c01170

L.K. designed and performed the study. Both authors wrote the paper. T.V. oversaw the project.

6

JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

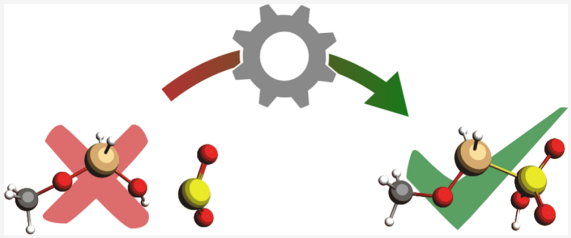
Improving the Silicon Interactions of GFN-xTB

Leonid Komissarov and Toon Verstraelen*

Cite This: https://doi.org/10.1021/acs.jcim.1c01170 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: A general-purpose density functional tight binding method, the GFN-xTB model is gaining increased popularity in accurate simulations that are out of scope for conventional *ab initio* formalisms. We show that in its original GFN1-xTB parametrization, organosilicon compounds are described poorly. This issue is addressed by re-fitting the model's silicon parameters to a data set of 10 000 reference compounds, geometry-optimized with the revPBE functional. The resulting GFN1(Si)-xTB parametrization shows improved accuracy in the prediction of system energies, nuclear forces, and geometries and should be considered for all applications of the GFN-xTB Hamiltonian to systems that contain silicon.



BACKGROUND AND SUMMARY

Silicon is the second most abundant element on earth.[1] Its physical properties make it a crucial building block for applications in electronics[2−4] and materials science.[5−7] More recently, organosilicon compounds have become of increasing interest to the field of organic synthesis in the role of (selective) intermediates[8−10] or catalysts.[11,12] One way scientists can study any of the aforementioned applications is through computational simulations of appropriate systems on modern computer hardware. Compared to experimental laboratory work, this domain of computational chemistry offers a high amount of flexibility when it comes to throughput, man-hours, and overall cost. A myriad of models can be employed to simulate a system of interest—too many to give a comprehensive overview of all of them here (we refer the reader to refs 13, 14 for a comprehensive introduction to the topic). Instead, we roughly classify two fundamental model types: *ab initio* and *empirical*. The former describes a model that is derived from the fundamental laws of physics and, without introducing approximations fitted to (experimental) reference data. Examples of *ab initio* models are the Hartree−Fock (HF) formalism, configuration interaction (CI) methods, and density functional theory (DFT) with nonempirical functionals like Perdew−Burke−Ernzerhof (PBE).[15] Explicit treatment of the electronic structure means that *ab initio* computations can be highly accurate but computationally slow and only limited to small systems of roughly tens to hundreds of atoms. Empirical models in contrast are fast and can handle system sizes of up to millions of atoms. This speedup is achieved through simplifications in the description of interatomic interactions. Examples of empirical models are classical force fields (FF) or machine learning potentials (MLP). One benefit of empirical models is the ability to fit their parameters to a specific chemical space. This allows for an improvement in prediction accuracy without the need to switch to a computationally more expensive model. The semiempirical GFN1-xTB[16] formalism falls under this category. The model has found a broad application in the modeling of small-to-medium organic molecules, where it is predominantly used for geometry optimizations.[17−20] Despite its success, we have observed large discrepancies between GFN1-xTB and DFT when comparing relative energies, optimized geometries, and nuclear gradients of organosilicon compounds. This issue is addressed by fitting the silicon parameters of GFN1-xTB to higher-level DFT data. We describe a reference data set of 10 000 organosilicon compounds, followed by the parameter optimization procedure in the Methods section. The Results and Discussion section discusses the shortcomings of the original GFN1-xTB model in more detail and compares them to our newly obtained parameters as well as to similar models from the literature. In contrast to the original parametrization, our parameters reproduce system energies and geometries more accurately, without compromise to the prediction accuracy of nonsilicon organic compounds, as tested on a subset of the ANI-1x[21] data.

METHODS

Organosilicon Reference Data. Initial structures for our reference data set are taken from the PubChem library.[22,23] The search query included the following filters:

- Heavy atom count between 1 and 15.

Received: September 24, 2021

   https://doi.org/10.1021/acs.jcim.1c01170
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

Journal of Chemical Information and Modeling pubs.acs.org/jcim Article

- Compound contains Si, O, C.
- Covalent unit count of 1.
- Molecular weight less than or equal to 200 g/mol.
- Total formal charge of 0.

From the total of 50 750 compounds matching the query, a random subset of 10k structures is selected. The resulting set has a mean heavy atom count of 10.5 (smallest: 3, largest: 13, standard deviation: 1.9). In addition to the chemical elements defined in the search query, H, N, Cl, S, F, P, and Br are included in the set. Elemental occurrences are presented in Table 1, with

Table 1. Elemental Occurrence in the Reference Data Set[a]

element	occurrence
C	10 000
O	10 000
Si	10 000
H	9994
N	3252
Cl	573
S	461
F	295
P	85
Br	27
1 Si	9318
2 Si	613
3 Si	59
4 Si	9
5 Si	1

[a]Structures containing at least one atom of the listed element are counted in the upper part. Structures with the exact number of silicon atoms are counted in the lower part.

nitrogen being the most and bromine the least prominent among the additional elements (other than Si, O, and C) in the set. We verify that the selected subset of 10k geometries is representative for all compounds matching the query by performing a t-distributed stochastic neighbor embedding[24] (t-SNE). Due to their ability to capture structural information, Morgan fingerprints[25,26] (as implemented in the RDKit[27] package) with a radius of 2 and length of 512 are used as input for the embedding. The scikit-learn[28] package is used for embedding into two dimensions. Results are presented in Figure 1. Compounds that are part of the randomly selected subset are represented by red dots, whereas all remaining ones are marked in blue. The embedding shows that the subset is a good representation of the complete search query, covering most of the embedded space. All structures from the selected subset are geometry-optimized with the Amsterdam Density Functional[29] (ADF) molecular simulation package, as integrated in the Amsterdam Modeling Suite.[30] We use the revPBE functional,[31,32] a "small" frozen core and the double-ζ polarized (DZP) basis set. Geometry convergence criteria are left at their default values, namely, 0.001 Hartree/Å, 0.00001 Hartree/Atom, and 0.1 Å for atomic gradients, energy, and atomic displacements, respectively. A quasi-Newton optimizer[33] in the delocalized coordinates space is used for the optimizations. Distributions of all of the convergence criteria at each structure's last optimization step are provided in Figure S1. Training and validation data sets are constructed from (1) atomic forces at the initial (un-optimized) geometries, considering Si atoms only, (2) energy differences between the same compound's initial and

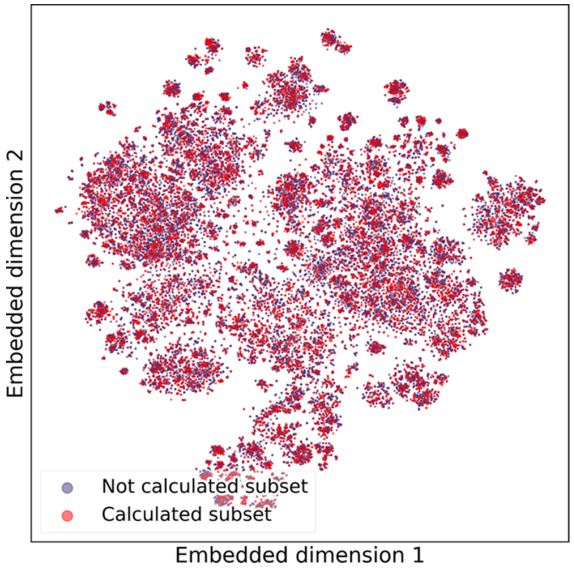


Figure 1. t-SNE[24] representation of all matching compounds from the PubChem[22,23] query based on Morgan fingerprints.[25,26] Each point represents one compound. Compounds that were selected for the reference calculations are marked in red, and the remaining in blue.

optimized geometries, and (3) root-mean-square deviation (RMSD) of the optimized geometries. Prior to the construction of the data sets, improbable outlier geometries are filtered by excluding 123 systems with an energy difference larger than 5 kJ/mol per atom between optimized and initial geometries.

Optimization of the Silicon GFN-xTB Parameters. Developed by Grimme et al., the GFN-xTB (also GFN1-xTB) model is a semiempirical method for the computation of a chemical system's Hamiltonian.[16] The method follows a density functional tight binding (DFTB) approximation, which describes the electronic energy E_{el} of a molecule as a function of its (valence) electron density $\rho(\boldsymbol{r}) = \rho_0(\boldsymbol{r}) + \delta_\rho(\boldsymbol{r})$, where $\boldsymbol{r}$ is a spatial coordinate.[34−36] The reference density ρ_0 is typically a superposition of individual atomic contributions, whereas $\delta\rho$ is the consequence chemical bonding and is assumed to be relatively small. In the DFTB formalism, the energy is approximated by a Taylor series up to the third order in $\delta\rho$ (corresponding to DFTB3) such that

$$E_{el}[\rho] = E^0[\rho_0] + E^1[\rho_0, (\delta\rho)^1] + E^2[\rho_0, (\delta\rho)^2] + E^3[\rho_0, (\delta\rho)^3] \quad (1)$$

The total GFN1-xTB energy is divided into electronic (el), repulsive (rep), dispersion (disp), and halogen-bonding (xb) interactions and can be written as

$$E = E_{el} + E_{rep} + E_{disp} + E_{xb} \quad (2)$$

Reference 16 describes the GFN-xTB Hamiltonian in more detail.

The parameter optimization is performed with the ParAMS parameter fitting package.[37] Starting from Grimme's original parametrization,[16] we optimize all 17 silicon parameters from the electronic and repulsive terms, plus one parameter specific to the Si−O atom pair for a total of 18 parameters. A summary of all optimized parameters and their values in the original parametrization are provided in Table 2 along with upper and lower

https://doi.org/10.1021/acs.jcim.1c01170
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

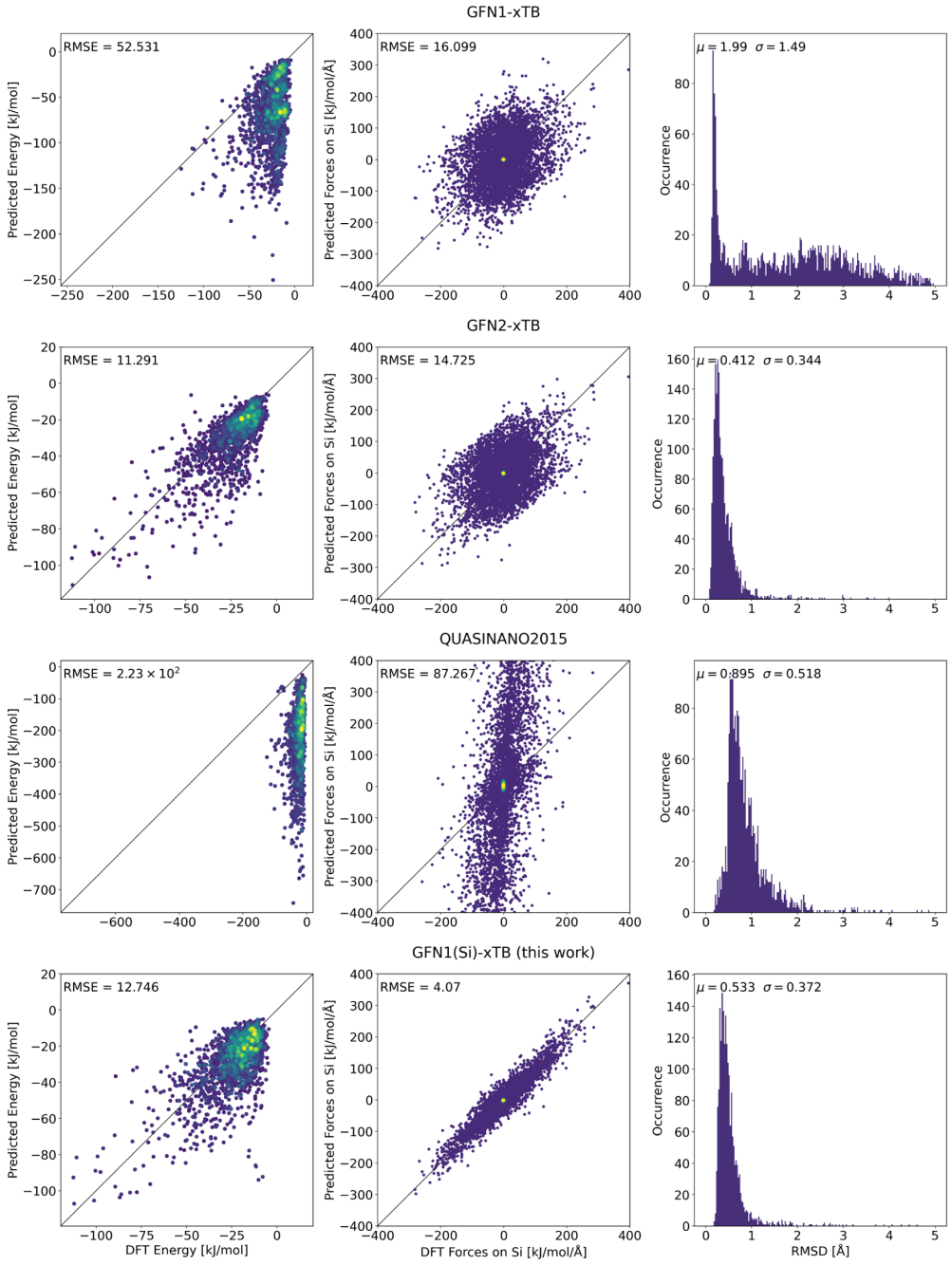


Figure 2. Validation set performance of various density functional tight binding models. Comparing GFN1-xTB,[16] GFN2-xTB,[40] QUASINANO2015,[41] and this GFN1(Si)-xTB (this work) from top to bottom. Columns from left to right depict energy differences, force components on the Si atoms, and RMSD of atomic positions (as described in the Methods section). *X* and *Y* values in the first two columns are reference properties and their DFTB predictions, respectively. Areas of lower point densities are depicted in dark blue, and higher densities in bright green. Root-mean-square error (RMSE) printed in the same units as the axes. Histograms in the right column show the RMSD between geometry-optimized reference and DFTB structures. Mean μ and standard deviation σ printed in ångström.

parameter ranges that were used during the optimization. Training and validation sets are created by randomly splitting the complete data set into relative sizes of 80 and 20%, respectively (splitting is based on unique structures, not their properties). Within the training set, each entry of relative energies, atomic forces, and RMSD is assigned a weight of 2.4, 28.0, and 1.0, respectively. The weights were determined by minimizing the standard deviation of all per-entry contributions

https://doi.org/10.1021/acs.jcim.1c01170
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

to the overall root-mean-square error (RMSE), as calculated with the initial GFN1-xTB parameters. Covariance matrix adaptation evolution strategy (CMA-ES),[38,39] a gradient-free, population-based optimization algorithm is used to optimize the GFN-xTB parameters. Population size and initial sampling width σ are set to 12 and 0.2, respectively. The optimization is set up to run for a maximum of 72 h. To prevent waste of computation time when stuck in local minima, an early stopping algorithm is set up to abort the optimization if there is no improvement in the training set loss after 1000 evaluations. At every optimization step, only a batch of 800 randomly selected jobs is computed to speed up convergence. To compensate for both, the noise introduced through the aforementioned batching and the pseudo-random sampling of CMA-ES, eight independent optimizations are performed. The best parameter set is selected based on the lowest training set loss function value.

RESULTS AND DISCUSSION

Performance of Models from the Literature. To serve as a baseline, the performance of three DFTB parametrizations from the literature is compared on our validation set. The first three rows of Figure 2 compare three general-purpose models, namely, GFN1-xTB[16] (the initial parameter set used for our optimizations), GFN2-xTB,[40] and the QUASINANO2015[41] set. Overall, GFN2-xTB (Figure 2 second row) is the most accurate of the three, with the lowest errors in the energy differences and the RMSD of geometry-optimized structures. Both GFN models predict atomic forces with the same accuracy (Figure 2 middle column, first and second rows). Although the QUASINANO2015 parameters are the least accurate of the three when comparing relative energies and atomic forces, distributions of the RMSD are better than for GFN1-xTB. We found that in most cases, the poor RMSD of GFN1-xTB can be traced back to unrealistic Si−O−R bond angles. This problem is visualized in Figure 3, showing the distributions of all Si−O−C angles for each of the aforementioned models. Note how DFT predicts an average angle of 121° (Figure 3a), while most angles computed with GFN1-xTB are almost colinear with an angle close to 180° (Figure 3b). This issue is not observed for the GFN2-xTB and QUASINANO2015 models (Figure 3c,d). In more extreme cases, geometry optimizations with GFN1-xTB resulted in rearrangements and bond dissociations. One example is presented in Figure 4, showing the optimized structure of sulfosilyloxymethane, as computed with revPBE and GFN1-xTB.

Optimized Silicon Parameters for GFN1-xTB. We report all optimized parameter values in Table 2. Following the previous section, we compare our new set of parameters, which we refer to as GFN1(Si)-xTB, in the last row of Figures 2 and 3e. Substantial improvements can be observed in all three of the fitted properties, bringing GFN1(Si)-xTB to an accuracy level that is comparable to GFN2-xTB. At the same time, calculations with the GFN1-xTB Hamiltonian have shown to be roughly 35% faster compared to the GFN2-xTB model. This has been tested by randomly selecting a batch of 200 geometries from our data set and measuring the time it took both models to calculate energies and atomic forces for all structures. The setup has been repeated 100 times to produce average timings.

A sanity check is performed by calculating the atomic gradients on a separate test set. We use the ANI-1x reference data by Smith et al.[21] for this purpose, which includes atomic gradients calculated with the $\omega B97X$[42] functional. The test set is constructed from one randomly selected conformation for each of the 3113 unique configurations in the ANI-1x data. Atomic gradients are computed with the GFN1-xTB and GFN1(Si)-xTB parameters. As expected, both parametrizations predict the same forces for the test set since the ANI-1x data does not

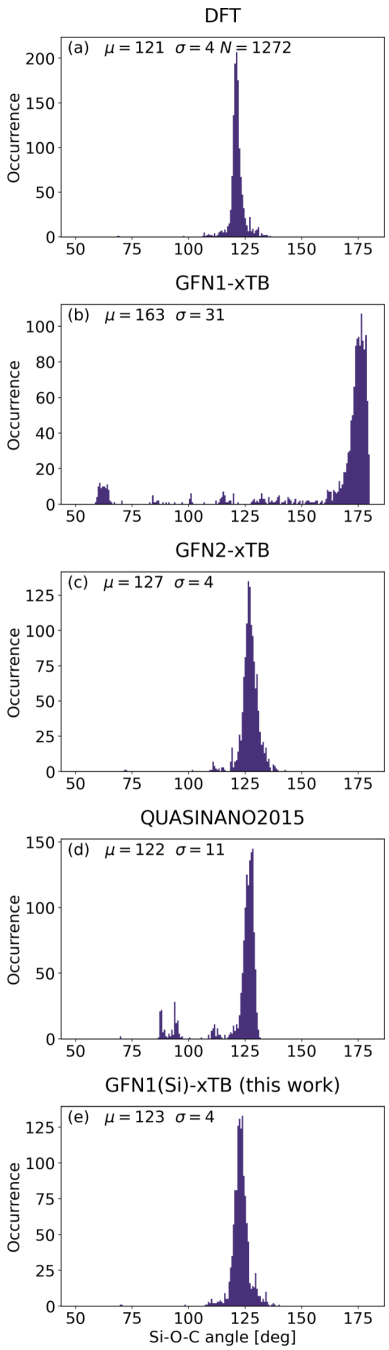


Figure 3. Si−O−C angle distributions of all optimized structures in the validation set. (a−c) Comparison of revPBE,[32] GFN1-xTB,[16] GFN2-xTB,[40] QUASINANO2015,[41] and this GFN1(Si)-xTB (this work), respectively. Printed text shows mean (μ), standard deviation (σ), and the total number of points (N).

https://doi.org/10.1021/acs.jcim.1c01170
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

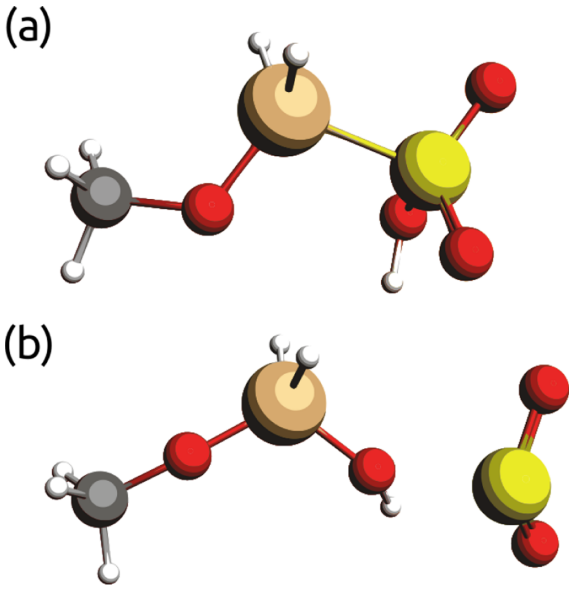


Figure 4. Optimized geometries of sulfosilyloxymethane (PubChem ID 154240983). Structures optimized with (a) revPBE and (b) GFN1-xTB. The latter shows a rearrangement and Si−O−R angles close to 180°. Hydrogen, carbon, oxygen, silicon, and sulfur are depicted in white, gray, red, beige, and yellow, respectively.

Table 2. List of All Optimized Parameter Names (Si Only Where Applicable, Atomic Subscript Dropped), Their Initial Values, and the Equation Number That Lists the Parameter as Described in Ref 16, Followed by the Optimization Bounds Used in This Work and the Optimized Values

parameter	equation	original value	optimization bounds	optimized value
η	5	+0.438	(+0.100, +3.500)	+2.251
Γ	9	+1.500	(+1.000, +2.000)	+1.125
α	13	+0.948	(+0.500, +1.800)	+1.036
Z	13	+16.898	(+8.000, +25.000)	+21.357
EN	10	+1.900	(+1.710, +2.090)	+2.089
K_{SiO}	10	+1.000	(+0.900, +1.100)	+0.969
3s level				
k_l^{poly}	11	−14.202	(-24.000, +15.000)	+14.825
κ^l	5	+0.000	(-10.000, +10.000)	−4.972
H^l	12	−14.506	(-30.000, -2.000)	−20.800
ζ_l	7	+1.522	(+0.400, +4.000)	+2.337
3p level				
k_l^{poly}	11	−3.893	(-10.000, +30.000)	−8.629
κ^l	5	−5.926	(-10.926, -0.926)	−6.046
H^l	12	−7.557	(-17.557, +2.443)	−3.526
ζ_l	7	+1.609	(+0.400, +4.000)	+1.576
3d level				
k_l^{poly}	11	+25.499	(-30.000, +44.000)	+36.572
κ^l	5	+0.000	(-10.000, +10.000)	+6.072
H^l	12	−2.508	(-12.508, +7.492)	−3.321
ζ_l	7	+1.169	(+0.400, +4.000)	+2.474

include any silicon atoms. A correlation plot between the $\omega B97X$ and GFN1-xTB forces is presented in Figure S2. For the ANI-1x set, the RMSE of the atomic gradients is roughly 28.5 kJ/mol/Å. This leads us to the conclusion that the GFN1(Si)-xTB parametrization can be used without any compromises when computing energies, gradients, and geometries of isolated organic compounds.

At this point, we would like to stress that the GFN1(Si)-xTB parameter set has been fitted to geometries, nuclear gradients, and energies, and as such should only be used for applications related to these properties. This forms a contrast to the original claim of the GFN model—the ability to represent **G**eometries, **F**requencies, and **N**oncovalent interactions accurately, as the latter two properties have neither been investigated nor fitted here. Although this calls for a future investigation, we argue that a first focus on corrected geometries, forces, and energies is most logical since (1) optimized geometries are the most popular prediction target when using GFN1-xTB[17−20] and (2) we expect that predictions of frequencies and noncovalent interactions on systems with poorly described geometries and energies will be of limited interest. A preliminary evaluation of a small subset shows that, compared to the original parametrization, the prediction accuracy of frequencies and dipole moments is not significantly affected by the new parameters. We also point out that the parametrization workflow presented here is fitting a dispersion-corrected model (GFN1) to reference data that has been computed without dispersion corrections (revPBE). This, in principle, introduces a systematic error to the training. However, we argue that in the case of a data set that consists of small molecules, errors due to dispersion correction will be negligible compared to the overall error of the GFN1 Hamiltonian. This has been verified by computing the same correlation plots as in Figure 2, but with dispersion corrections explicitly turned off for the GFN1 parametrizations. The results are presented in Figure S3, showing that dispersion corrections have a minimal effect on the predicted properties.

SUMMARY AND OUTLOOK

We have introduced improved silicon parameters for the GFN1-xTB Hamiltonian, addressing the issue of poor performance in the prediction of energies, atomic gradients, and geometries. The improved parameter set, titled GFN1(Si)-xTB, was obtained through an easy-to-reproduce fitting scheme based on an *ab initio* data set of 10 000 small organosilicon compounds and is able to predict the aforementioned properties significantly more accurately. We hope that the published workflow will encourage future research in the field of parameter optimization. Moreover, the publication of the complete reference data set, consisting of optimization trajectories, energies, and gradients, will be of general use to the scientific community. Future developments will focus on the effective inclusion of frequencies and noncovalent interactions in the training as well as more application-driven parametrizations of more complex structures.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.1c01170.

Distributions of convergence criteria for the organosilicon reference data set, non-D3-corrected correlation plots for both GFN1 parametrizations, documentation of the stored data format, and correlation plot comparing the atomic forces on the ANI-1x subset (PDF)

https://doi.org/10.1021/acs.jcim.1c01170
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

6

AUTHOR INFORMATION

Corresponding Author

Toon Verstraelen − *Center for Molecular Modeling (CMM), Ghent University, B-9052 Ghent, Belgium;* orcid.org/0000-0001-9288-5608; Email: toon.verstraelen@ugent.be

Author

Leonid Komissarov − *Center for Molecular Modeling (CMM), Ghent University, B-9052 Ghent, Belgium;* orcid.org/0000-0001-6011-1632

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.1c01170

Author Contributions

L.K. designed and performed the study. Both authors wrote the manuscript. T.V. oversaw the project.

Notes

The authors declare no competing financial interest.

GFN1(Si)-xTB parameters (in the AMS format), reference data, including training and validation sets, and files needed to run the parameter optimization scheme are available at DOI: 10.24435/materialscloud:14-4m or the Materials Cloud Archive record 2021.152. See the Supporting Information or data repository for a description of the reference data formats. For the ANI-1x test set, refer to ref 21. The Amsterdam Modeling Suite[30] (v. 2020.203), which includes the ParAMS package[37] (v. 0.5.1), is a commercial software, for which a free trial may be requested at www.scm.com. GFN2-xTB calculations were performed with the openly available xtb-python package[43] (v. 20.1).

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 814143. T.V. acknowledges funding of the research board of Ghent University. The computational resources (Stevin Supercomputer Infrastructure) and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by Ghent University, FWO, and the Flemish Government—department EWI.

REFERENCES

(1) Rumble, J. *CRC Handbook of Chemistry and Physics: A Ready-Reference Book of Chemical and Physical Data*; CRC Press/Taylor & Francis Group, 2020.

(2) McConnell, H. M.; Owicki, J. C.; Parce, J. W.; Miller, D.; Baxter, G.; Wada, H. G.; Pitchford, S. The Cytosensor Microphysiometer: Biological Applications of Silicon Technology. *Science* **1992**, *257*, 1906−1912.

(3) Bruel, M. Silicon on Insulator Material Technology. *Electron. Lett.* **1995**, *31*, 1201−1202.

(4) Chapin, D. M.; Fuller, C. S.; Pearson, G. L. A New Silicon p-n Junction Photocell for Converting Solar Radiation Into Electrical Power. *J. Appl. Phys.* **1954**, *25*, 676−677.

(5) Baerlocher, C.; McCusker, L. B.; Olson, D. H. *Atlas of Zeolite FrameworkTypes*; Elsevier, 2007.

(6) Park, J.-H.; Gu, L.; Von Maltzahn, G.; Ruoslahti, E.; Bhatia, S. N.; Sailor, M. J. Biodegradable Luminescent Porous Silicon Nanoparticles for in Vivo Applications. *Nat. Mater.* **2009**, *8*, 331−336.

(7) Yang, P.; Gai, S.; Lin, J. Functionalized Mesoporous Silica Materials for Controlled Drug Delivery. *Chem. Soc. Rev.* **2012**, *41*, 3679−3698.

(8) Zhang, H.-J.; Priebbenow, D. L.; Bolm, C. Acylsilanes: Valuable Organosilicon Reagents in Organic Synthesis. *Chem. Soc. Rev.* **2013**, *42*, 8540−8571.

(9) Lalonde, M.; Chan, T. Use of Organosilicon Reagents as Protective Groups in Organic Synthesis. *Synthesis* **1985**, *1985*, 817−845.

(10) Hatanaka, Y.; Hiyama, T. Highly Selective Cross-Coupling Reactions of Organosilicon Compounds Mediated by Fluoride Ion and a Palladium Catalyst. *Synlett* **1991**, *1991*, 845−853.

(11) dos Santos, J. H. Z.; Greco, P. P.; Stedile, F. C.; Dupont, J. Organosilicon-Modified Silicas as Support for Zirconocene Catalyst. *J. Mol. Catal. A: Chem.* **2000**, *154*, 103−113.

(12) Walker, J. C. L.; Klare, H. F. T.; Oestreich, M. Cationic Silicon Lewis Acids in Catalysis. *Nat. Rev. Chem.* **2020**, *4*, 54−62.

(13) Cramer, C. *Essentials of Computational Chemistry: Theories and Models*; Wiley: Chichester, West Sussex, England, Hoboken, NJ, 2004.

(14) Frenkel, D. *Understanding Molecular Simulation: From Algorithms to Applications*; Academic Press: San Diego, 2002.

(15) Perdew, J.; Burke, K.; Ernzerhof, M. Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.* **1996**, *77*, 3865−3868.

(16) 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.* **2017**, *13*, 1989−2009.

(17) Cavasin, A. T.; Hillisch, A.; Uellendahl, F.; Schneckener, S.; Göller, A. H. Reliable and Performant Identification of Low-Energy Conformers in the Gas Phase and Water. *J. Chem. Inf. Model.* **2018**, *58*, 1005−1020.

(18) Hahn, R.; Bohle, F.; Fang, W.; Walther, A.; Grimme, S.; Esser, B. Raising the Bar in Aromatic Donor−Acceptor Interactions with Cyclic Trinuclear Gold(I) Complexes as Strong π-Donors. *J. Am. Chem. Soc.* **2018**, *140*, 17932−17944.

(19) Casajus, H.; Dubreucq, E.; Tranchimand, S.; Perrier, V.; Nugier-Chauvin, C.; Cammas-Marion, S. Lipase-Catalyzed Ring-Opening Polymerization of Benzyl Malolactonate: An Unusual Mechanism? *Biomacromolecules* **2020**, *21*, 2874−2883.

(20) Backhouse, O. J.; Santana-Bonilla, A.; Booth, G. H. Scalable and Predictive Spectra of Correlated Molecules with Moment Truncated Iterated Perturbation Theory. *J. Phys. Chem. Lett.* **2021**, *12*, 7650−7658.

(21) Smith, J. S.; Zubatyuk, R.; Nebgen, B.; Lubbers, N.; Barros, K.; Roitberg, A. E.; Isayev, O.; Tretiak, S. The ANI-1ccx and ANI-1x Data Sets, Coupled-Cluster and Density Functional Theory Properties for Molecules. *Sci. Data* **2020**, *7*, No. 134.

(22) Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; Bolton, E. E. PubChem 2019 Update: Improved Access to Chemical Data. *Nucleic Acids Res.* **2018**, *47*, D1102−D1109.

(23) Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; Bolton, E. E. PubChem in 2021: new data content and improved web interfaces. *Nucleic Acids Res.* **2021**, *49*, D1388−D1395.

(24) Van der Maaten, L.; Hinton, G. Visualizing Data Using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579−2605.

(25) Morgan, H. L. The Generation of a Unique Machine Description for Chemical Structures - A Technique Developed at Chemical Abstracts Service. *J. Chem. Doc.* **1965**, *5*, 107−113.

(26) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50*, 742−754.

(27) Landrum, G. RDKit: Open-Source Cheminformatics, 2021. http://www.rdkit.org.

(28) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.

(29) te Velde, G.; Bickelhaupt, F. M.; Baerends, E. J.; Guerra, C. F.; van Gisbergen, S. J. A.; Snijders, J. G.; Ziegler, T. Chemistry with ADF. *J. Comput. Chem.* **2001**, *22*, 931−967.

(30) Rüger, R.; Franchini, M.; Trnka, T.; Yakovlev, A.; van Lenthe, E.; Philipsen, P.; van Vuren, T.; Klumpers, B.; Soini, T. Amsterdam Modeling Suite, 2019. https://scm.com.

https://doi.org/10.1021/acs.jcim.1c01170
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

(31) Perdew, J. P.; Burke, K.; Ernzerhof, M. Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.* **1996**, *77*, 3865−3868.
(32) Zhang, Y.; Yang, W. Comment on "Generalized Gradient Approximation Made Simple". *Phys. Rev. Lett.* **1998**, *80*, 890.
(33) Swart, M.; Bickelhaupt, F. M. Optimization of Strong and Weak Coordinates. *Int. J. Quantum Chem.* **2006**, *106*, 2536−2544.
(34) Elstner, M.; Seifert, G. Density Functional Tight Binding. *Philos. Trans. R. Soc., A* **2014**, *372*, No. 20120483.
(35) Christensen, A. S.; Kubař, T.; Cui, Q.; Elstner, M. Semiempirical Quantum Mechanical Methods for Noncovalent Interactions for Chemical and Biochemical Applications. *Chem. Rev.* **2016**, *116*, 5301−5337.
(36) Hourahine, B.; Aradi, B.; Blum, V.; Bonafé, F.; Buccheri, A.; Camacho, C.; Cevallos, C.; Deshaye, M. Y.; Dumitrică, T.; Dominguez, A.; Ehlert, S.; Elstner, M.; van der Heide, T.; Hermann, J.; Irle, S.; Kranz, J. J.; Köhler, C.; Kowalczyk, T.; Kubař, T.; Lee, I. S.; Lutsker, V.; Maurer, R. J.; Min, S. K.; Mitchell, I.; Negre, C.; Niehaus, T. A.; Niklasson, A. M. N.; Page, A. J.; Pecchia, A.; Penazzi, G.; Persson, M. P.; Řezáč, J.; Sánchez, C. G.; Sternberg, M.; Stöhr, M.; Stuckenberg, F.; Tkatchenko, A.; Yu, V. W.; Frauenheim, T. DFTB+, a Software Package for Efficient Approximate Density Functional Theory Based Atomistic Simulations. *J. Chem. Phys.* **2020**, *152*, No. 124101.
(37) Komissarov, L.; Rüger, R.; Hellström, M.; Verstraelen, T. ParAMS: Parameter Optimization for Atomistic and Molecular Simulations. *J. Chem. Inf. Model.* **2021**, *61*, 3737−3743.
(38) Hansen, N.; Ostermeier, A. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evol. Comput.* **2001**, *9*, 159−195.
(39) Hansen, N.; Kern, S. In *Evaluating the CMA Evolution Strategy on Multimodal Test Functions*, International Conference on Parallel Problem Solving from Nature; Springer: Berlin, Heidelberg, 2004; pp 282−291.
(40) Bannwarth, C.; Ehlert, S.; Grimme, S. GFN2-xTB—An Accurate and Broadly Parametrized Self-Consistent Tight-Binding Quantum Chemical Method with Multipole Electrostatics and Density-Dependent Dispersion Contributions. *J. Chem. Theory Comput.* **2019**, *15*, 1652−1671.
(41) Oliveira, A. F.; Philipsen, P.; Heine, T. DFTB Parameters for the Periodic Table, Part 2: Energies and Energy Gradients from Hydrogen to Calcium. *J. Chem. Theory Comput.* **2015**, *11*, 5209−5218.
(42) Chai, J.-D.; Head-Gordon, M. Long-Range Corrected Hybrid Density Functionals with Damped Atom-Atom Dispersion Corrections. *Phys. Chem. Chem. Phys.* **2008**, *10*, 6615−6620.
(43) Python API for the Extended Tight Binding Program, 2021. https://github.com/grimme-lab/xtb-python.

https://doi.org/10.1021/acs.jcim.1c01170
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

6

Zeo-1, a computational data set of zeolite structures

L. Komissarov and T. Verstraelen

Sci. Data **2022**, *9*, 61
DOI: 10.1038/s41597-022-01160-5

L.K. designed and performed the study. Both authors wrote the paper. T.V. oversaw the project.

6

scientific data

OPEN

DATA DESCRIPTOR

Zeo-1, a computational data set of zeolite structures

Leonid Komissarov & Toon Verstraelen

Fast, empirical potentials are gaining increased popularity in the computational fields of materials science, physics and chemistry. With it, there is a rising demand for high-quality reference data for the training and validation of such models. In contrast to research that is mainly focused on small organic molecules, this work presents a data set of geometry-optimized bulk phase zeolite structures. Covering a majority of framework types from the Database of Zeolite Structures, this set includes over thirty thousand geometries. Calculated properties include system energies, nuclear gradients and stress tensors at each point, making the data suitable for model development, validation or referencing applications focused on periodic silica systems.

Background & Summary

Atomistic models are an essential tool for the prediction of thermodynamic, mechanical or biochemical properties of a substance. More recently, the use of pre-trained models has become increasingly popular due to their comparably low complexity and high accuracy on modern hardware[1–6]. In order for such models to perform well, their empirical parameters require fitting to high-quality reference data. Depending on the application, reference data are either experimental, or come from computationally more expensive *ab initio* calculations. Although there are already a handful of large computational data sets covering small organic molecules[7–9], such data is still scarce for larger periodic systems (*cf*. Materials Cloud Archive[10,11] or the NOMAD database[12,13]). Motivated by this fact, we present a quantum-chemical data set for zeolites. Zeolites are porous materials comprised of interconnected SiO_4 or AlO_4 tetrahedra. Their properties can be fine-tuned through synthesis of materials with specific pore size, or the inclusion of additional metal cation sites[14–17]. Because of their topology and synthetic flexibility, zeolites have various applications as adsorbents[18–20] and catalysts[17,21–23]. To this day, a myriad of different zeolite framework types is available experimentally, and many more hypothetical structures can be derived[24–26]. The documentation of fundamental zeolite framework types and derived materials has led to the publication of the well-known *Atlas of Zeolite Structures*[27] in several editions. The atlas lists each unique framework type by its three-letter-code, as assigned by the by the Structure Commission of the International Zeolite Association (IZA). Today, its contents are available online at the *Database of Zeolite Structures*[28], which we use as a source of initial structures for our data set. In this first installment, we include properties for 204 out of the currently available 256 zeolite framework types in the database (a total of 226 unique geometries when also considering derived materials). Our descriptor provides the complete optimization trajectories for each system with atomic positions, lattice vectors, atomic gradients and stress tensors at each step. We envision future extensions of the data set to focus on derived geometries, covering structural defects and host-guest interactions.

Methods

Initial zeolite structures are collected from the public *Database of Zeolite Structures*[28] in the *Crystallographic Information File* (CIF) format, before conversion to the XYZ format with the Atomic Simulation Environment[29] (ASE) package. After selection of all systems with less than 301 atoms, each is manually filtered by removing redundant atom positions in case of fractional occupancies and adding missing hydrogen atoms where needed. Each structure's coordinates and cell parameters are energy-minimized with the periodic density functional code BAND[30], as implemented in the Amsterdam Modeling Suite[31] (AMS). The calculations are performed with the revPBE functional[32,33], a 'Small' frozen core and the double-ζ polarized (DZP) basis set. Grimme's D3(BJ) dispersion correction[34] is applied to all calculations. Previous research has shown that the selected level of theory can accurately reproduce zeolite geometries, albeit slightly overestimating the Si-O bond length (in the range of 2 pm) and smaller Si-O-X angles (in the range of 5 degrees) when compared to experimental results[35,36]. At the same time, dispersion-corrected functionals are generally more accurate when describing adsorption

Center for Molecular Modeling (CMM), Ghent University, Technologiepark-Zwijnaarde 46, B-9052, Ghent, Belgium. e-mail: leonid.komissarov@ugent.be; toon.verstraelen@ugent.be

Data	Unit	Key	Array Shape
Atomic Numbers	—	`numbers`	$(R,)$
Atomic Coordinates	Å	`xyz`	$(N, R, 3)$
x-, y- and, z-Components of the Lattice Vectors	Å	`lattice`	$(N, 3, 3)$
Energy	hartree	`energy`	$(N,)$
Nuclear Gradients	hartree/bohr	`gradients`	$(N, R, 3)$
Stress Tensors	atomic units	`stress`	$(N, 3, 3)$
Hirshfeld Charges	atomic units	`charges`	$(R,)$

Table 1. Overview of the data structures stored in a .npz file. Each array can be accessed through the respective key. The variables N and R denote the number of geometry optimization steps and the system size respectively. Partial charges are only computed for the last geometry.

Element	Occurrence
Si	226
O	226
H	21
Al	12
N	4
Ca	4
Ge	3
Li	2
Na	2
K	2
C	2
F	1
Be	1
Cs	1
Ba	1

Table 2. Elemental occurrences in the complete data set. Counting all structures containing at least one atom of the listed element. Each element's isolated atomic energy is listed in hartree.

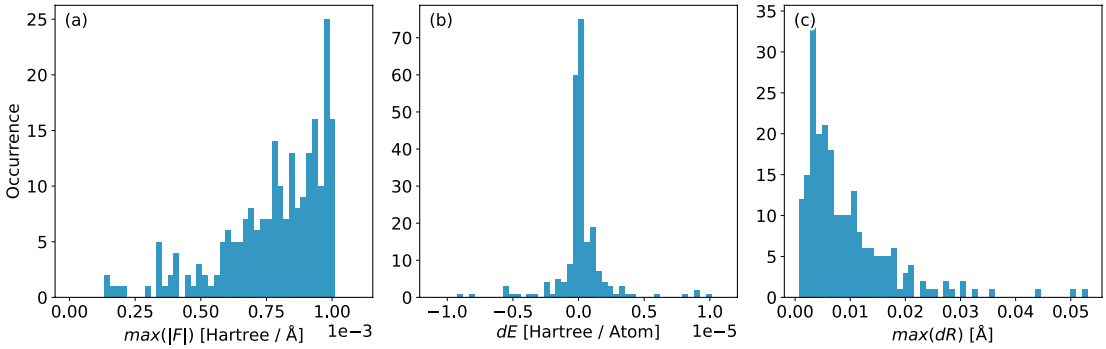


Fig. 1 Distribution of convergence criteria at the last optimization step for all calculated systems in the data set. Showing (**a**) the highest absolute component of all nuclear gradients, (**b**) change in system energy and (**c**) highest relative atomic displacement.

processes[37–39]. For the optimization of the initial structures, geometry convergence criteria are left at their default values, namely 0.001 Hartree/Å, 0.00001 Hartree/Atom and 0.1 Å for atomic gradients, energy and atomic displacements respectively. We use a Quasi-Newton optimizer[40] in the delocalized coordinates space for the initial optimizations. Cases of problematic convergence are restarted with the FIRE[41] optimizer.

Data Records

The data is made available at the Materials Cloud Archive[42]. Each system's trajectory is stored in an individual NumPy[43]. *npz* file. We describe the data types held in each file in Table 1, storing the complete geometry optimization trajectory, including atomic coordinates, system energies, nuclear gradients, lattice vectors and stress

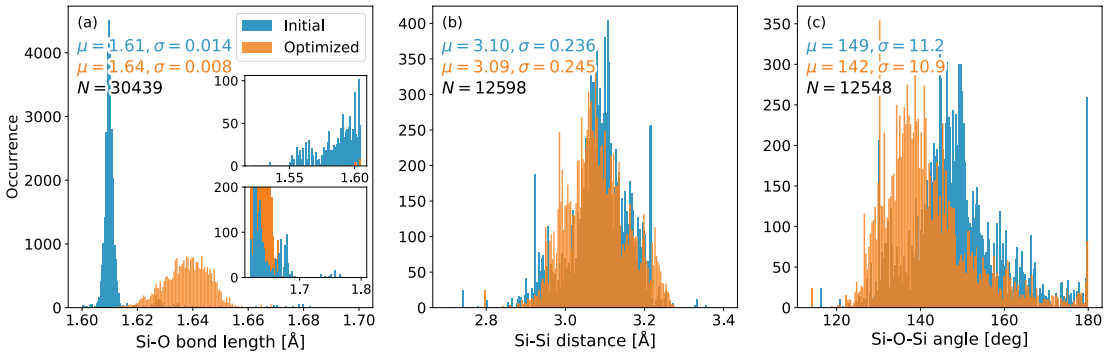


Fig. 2 Distributions of (**a**) Si-O bond lengths, (**b**) Si-Si distances in the second coordination sphere and (**b**) Si-O-Si angles as calculated from all geometries in the data set. Blue and orange bars denote data from initial and optimized geometries, respectively. Mean μ and standard deviation σ printed in the same color as the underlying data. N denotes the total sample size.

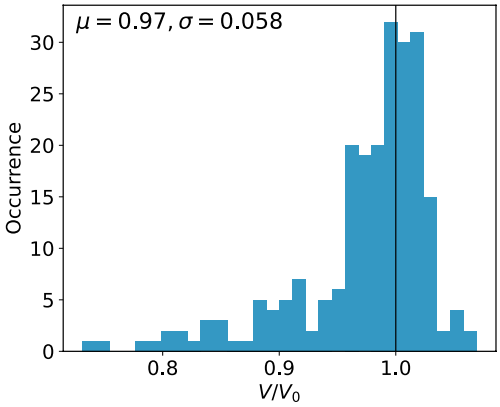


Fig. 3 Distribution of relative cell volumes per system as the quotient of optimized-to-initial cell volumes. Values below 1 describe a shrinking cell as the optimization progresses. Black line marks $V/V_0 = 1$. Sample size is 226.

Bond	Mean	Std. Dev.	Number of points
Si-O	1.638	0.0085	30439
H-O	0.999	0.1174	266
Al-O	1.763	0.0133	234
Ge-O	1.795	0.0239	202
Na-O	2.473	0.0841	104
C-C	1.540	0.0049	100
C-H	1.100	0.0027	98
K-O	3.175	0.4809	61
Ca-O	2.469	0.0925	57
N-H	1.055	0.0913	50
Si-K	3.945	0.3196	41
Cs-O	3.429	0.2820	28
Li-O	1.970	0.0263	21
Be-O	1.669	0.0152	16
Al-K	3.625	0.1650	14
C-N	1.472	0.0037	10
Ba-O	2.903	0.1261	10

Table 3. Mean atomic bond length distributions and their standard deviations (std. dev.) in in ångström. Averaged over all geometry-optimized structures.

Angle	Mean	Std. Dev.	Number of points
Si-O-Si	148.7	11.2	12548
Si-O-Al	140.6	8.9	170
Si-O-K	106.8	8.8	81
Si-O-Na	112.8	15.1	64
Si-O-Ge	143.2	12.0	52
Si-O-H	110.7	7.9	40
Si-O-Cs	101.6	6.9	36
Si-O-Ca	118.5	16.6	19
Si-O-Be	129.9	0.2	16
Si-O-Li	112.7	4.2	8
Si-O-Ba	112.6	14.1	5

Table 4. Mean Si-O-R angle distributions and their standard deviations (std. dev.) in degrees. Averaged over all geometry-optimized structures.

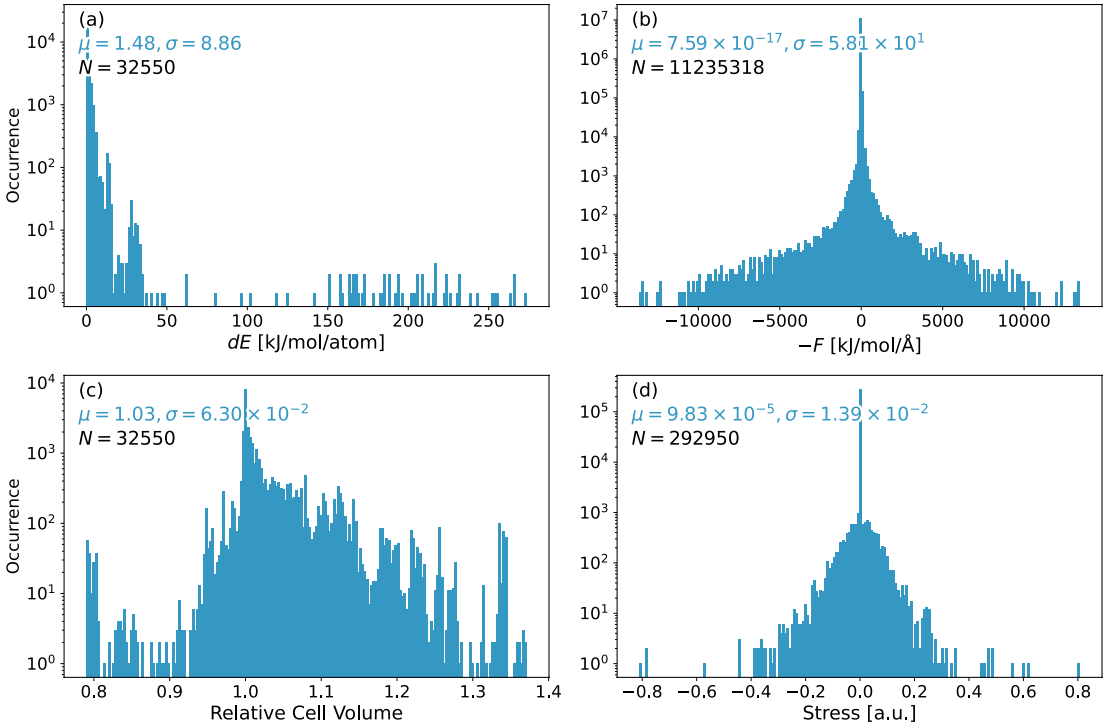


Fig. 4 Distributions of physical quantities in the data set. Showing (**a**) energy differences per atom, relative to the respective energy of the optimized system; (**b**) atomic gradient components; (**c**) unit cell volumes, relative to the optimized system's volume; (**d**) stress tensor components. Data is printed on a logarithmic y-scale for a clear display of the distribution. Mean μ and standard deviation σ printed in the same units as the underlying data. N denotes the total sample size.

tensors for each geometry optimization step. Entries at the first position correspond to the input structure; the last position holds the data for the final, optimized structure. Hirshfeld partial charges[44] are provided for the final (optimized) geometries. Atomic coordinates and lattice vectors are stored in ångström, all other properties are stored in atomic units.

Technical Validation

The complete data set includes geometry optimizations of 226 systems, resulting in a total of 32550 geometries. System sizes range between 15 and 334 atoms (mean: 126). We illustrate the convergence of all reference calculations in Fig. 1, showing that all optimized systems are well within the defined convergence criteria. Elemental occurrences in the data set are listed in Table 2. Si-O, Si-Si distances as well as Si-O-Si angles are presented in

Fig. 2 as the most prominent geometrical descriptors. As most of the initial structures from the IZA database are idealized geometries[45], a sharp mean for the Si-O bond distance can be observed at roughly 161 pm (Fig. 2a, blue histogram). Long tails in the distribution vanish and the mean is shifted towards approximately 164 pm when considering geometry-optimized structures (Fig. 2a, orange histogram). Considering the Si-O-Si angles, a slight shift towards smaller values is observed (mean of 149 vs. 142 degrees, Fig. 2c). Both effects have been previously reported by Fischer *et al.*[35,36] and are inherent to the selected level of theory. Distributions of the Si-Si distances in the second coordination sphere do not shift significantly when comparing initial and optimized geometries (Fig. 2b). Relative changes in the cell volumes are presented in Fig. 3 as the ratio of each system's optimized-to-initial volume. Values below 1 translate to a shrinking unit cell as the optimization progresses. Overall, the geometrical descriptors are in good agreement with experimental data[46–51]. Additional averages for bond distances and angles are summarized in Tables 3, 4 respectively. Distributions of energies, atomic gradients, cell volumes and stress tensors are depicted in Fig. 4. As expected from geometry optimization trajectories, all properties have – with the exception of relative cell volumes – a distinct mean close to zero. Structures close to the initial input geometries contribute to the relatively high standard deviations. Evaluation of the relative cell volumes shows a shifted distribution, with roughly 76% of all structures having a larger volume than their respective optimized geometry. A detailed overview of all calculated structures, sorted by their IZA three-letter-code, the system size and number of iterations is provided in Online Table 1.

Usage Notes

No data points were filtered as outliers with regards to the distributions of chemical properties (see. Figure 4). Consecutive structures from the same optimization trajectory will be autocorrelated. The data repository provides an interactive plotting script, displaying the system energy, maximum absolute component of the nuclear gradients and the cell volume at every iteration step for each structure. This requires the Bokeh[52] (v. 2.3.1) package for Python to be installed. SHA-1 hash sums are provided for each file to guarantee data integrity, as well as an example input script for a calculation with BAND. Naming conventions: Derived materials are referred to by their IZA three-letter-code, *e.g.* H-EU-12 is tabulated as ETL_0. Leading non-alphabetical characters have been removed, *e.g.* *-ITN is tabulated as ITN.

Code availability

Downloads of the Atomic Simulation Environment[29] (v. 3.21.1) and NumPy[43] (v. 1.20.1) packages for Python are freely available. Amsterdam Modeling Suite[31] (v. 2020.203, r92091) is a commercial software, for which a free trial may be requested at www.scm.com.

Received: 2 September 2021; Accepted: 14 January 2022;
Published online: 22 February 2022

References

1. Smith, J. S. *et al.* Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning. *Nature Communications* **10** (2019).
2. Bannwarth, C., Ehlert, S. & Grimme, S. GFN2-xTB—an accurate and broadly parametrized self-consistent tight-binding quantum chemical method with multipole electrostatics and density-dependent dispersion contributions. *J. Chem. Theory Comput.* **15**, 1652–1671 (2019).
3. Shao, Y., Hellström, M., Mitev, P. D., Knijff, L. & Zhang, C. PiNN: A python library for building atomic neural networks of molecules and materials. *Journal of Chemical Information and Modeling* **60**, 1184–1193 (2020).
4. Satorras, V. G., Hoogeboom, E. & Welling, M. E(n) equivariant graph neural networks. Preprint at https://arxiv.org/abs/2102.09844 (2021).
5. Behler, J. & Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **98**, 146401 (2007).
6. Kondratyuk, N. *et al.* Performance and scalability of materials science and machine learning codes on the state-of-art hybrid supercomputer architecture. In Voevodin, V. & Sobolev, S. (eds.) *Supercomputing*, 597–609 (Springer International Publishing, Cham, 2019).
7. Smith, J. S., Isayev, O. & Roitberg, A. E. ANI-1, a data set of 20 million calculated off-equilibrium conformations for organic molecules. *Scientific Data* **4**, 170193 (2017).
8. Smith, J. S. *et al.* The ANI-1ccx and ANI-1x data sets, coupled-cluster and density functional theory properties for molecules. *Scientific Data* **7**, 134 (2020).
9. Ramakrishnan, R., Dral, P. O., Rupp, M. & von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data* **1**, 140022 (2014).
10. *Materials Cloud Archive*. https://archive.materialscloud.org/ (2021).
11. Talirz, L. *et al.* Materials cloud, a platform for open computational science. *Scientific Data* **7**, 299 (2020).
12. *NOMAD Laboratory*. https://nomad-lab.eu/ (2021).
13. Draxl, C. & Scheffler, M. Nomad: The fair concept for big data-driven materials science. *MRS Bulletin* **43**, 676–682 (2018).
14. Davis, M. E. & Lobo, R. F. Zeolite and molecular sieve synthesis. *Chemistry of Materials* **4**, 756–768 (1992).
15. Cundy, C. S. Microwave techniques in the synthesis and modification of zeolite catalysts. a review. *Collection of Czechoslovak Chemical Communications* **63**, 1699–1723 (1998).
16. Chen, L.-H. *et al.* Hierarchically structured zeolites: synthesis, mass transport properties and applications. *Journal of Materials Chemistry* **22**, 17381 (2012).
17. Moliner, M., Martnez, C. & Corma, A. Multipore zeolites: Synthesis and catalytic applications. *Angewandte Chemie International Edition* **54**, 3560–3579 (2015).
18. Ozekmekci, M., Salkic, G. & Fellah, M. F. Use of zeolites for the removal of H2S: a mini-review. *Fuel Processing Technology* **139**, 49–60 (2015).
19. Papaioannou, D., Katsoulos, P., Panousis, N. & Karatzias, H. The role of natural and synthetic zeolites as feed additives on the prevention and/or the treatment of certain farm animal diseases: a review. *Microporous and Mesoporous Materials* **84**, 161–170 (2005).
20. Dehghan, R. & Anbia, M. Zeolites for adsorptive desulfurization from fuels: a review. *Fuel Processing Technology* **167**, 99–116 (2017).

21. Derouane, E. *et al.* The acidity of zeolites: concepts, measurements and relation to catalysis: A review on experimental and theoretical methods for the study of zeolite acidity. *Catalysis Reviews* **55**, 454–515 (2013).
22. Weitkamp, J. Zeolites and catalysis. *Solid State Ionics* **131**, 175–188 (2000).
23. Corma, A. State of the art and future challenges of zeolites as catalysts. *Journal of Catalysis* **216**, 298–312 (2003).
24. Treacy, M. M. J., Randall, K. H., Rao, S., Perry, J. A. & Chadi, D. J. Enumeration of periodic tetrahedral frameworks. *Zeitschrift für Kristallographie - Crystalline Materials* **212**, 768–791 (1997).
25. Treacy, M. M. J. & Foster, M. *Atlas of Prospective Zeolite Structures*. http://www.hypotheticalzeolites.net/ (2021).
26. Pophale, R., Cheeseman, P. A. & Deem, M. W. A database of new zeolite-like materials. *Phys. Chem. Chem. Phys.* **13**, 12407–12412 (2011).
27. Baerlocher, C., McCusker, L. & Olson, D. *Atlas of Zeolite Framework Types* (Published on behalf of the Structure Commission of the International Zeolite Association by Elsevier, 2007).
28. Baerlocher, C. & McCusker, L. *Database of Zeolite Structures*. http://www.iza-structure.org/databases/.
29. Larsen, A. H. *et al.* The atomic simulation environment—a python library for working with atoms. *Journal of Physics: Condensed Matter* **29**, 273002 (2017).
30. te Velde, G. & Baerends, E. J. Precise density-functional method for periodic structures. *Phys. Rev. B* **44**, 7888–7903 (1991).
31. Rüger *et al. Amsterdam Modeling Suite*. https://scm.com (2019).
32. Perdew, J. P., Burke, K. & Ernzerhof, M. Generalized gradient approximation made simple. *Physical Review Letters* **77**, 3865–3868 (1996).
33. Zhang, Y. & Yang, W. Comment on "generalized gradient approximation made simple". *Physical Review Letters* **80**, 890–890 (1998).
34. Grimme, S., Ehrlich, S. & Goerigk, L. Effect of the damping function in dispersion corrected density functional theory. *Journal of Computational Chemistry* **32**, 1456–1465 (2011).
35. Fischer, M., Evers, F. O., Formalik, F. & Olejniczak, A. Benchmarking dft-gga calculations for the structure optimisation of neutral-framework zeotypes. *Theoretical Chemistry Accounts* **135** (2016).
36. Fischer, M. & Angel, R. J. Accurate structures and energetics of neutral-framework zeotypes from dispersion-corrected dft calculations. *The Journal of Chemical Physics* **146**, 174111 (2017).
37. Göltl, F., Grüneis, A., Bučko, T. & Hafner, J. Van der waals interactions between hydrocarbon molecules and zeolites: periodic calculations at different levels of theory, from density functional theory to the random phase approximation and mÃ¸ller-plesset perturbation theory. *The Journal of Chemical Physics* **137**, 114111 (2012).
38. Rehak, F. R., Piccini, G., Alessio, M. & Sauer, J. Including dispersion in density functional theory for adsorption on flat oxide surfaces, in metal—organic frameworks and in acidic zeolites. *Physical Chemistry Chemical Physics* **22**, 7577–7585 (2020).
39. Stanciakova, K., Louwen, J. N., Weckhuysen, B. M., Bulo, R. E. & Göltl, F. Understanding water—zeolite interactions: on the accuracy of density functionals. *The Journal of Physical Chemistry C* **125**, 20261–20274 (2021).
40. Swart, M. & Bickelhaupt, F. M. Optimization of strong and weak coordinates. *International Journal of Quantum Chemistry* **106**, 2536–2544 (2006).
41. Bitzek, E., Koskinen, P., Gähler, F., Moseler, M. & Gumbsch, P. Structural relaxation made simple. *Physical Review Letters* **97** (2006).
42. Komissarov, L. & Verstraelen, T. *Zeo-1: a computational data set of zeolite structures*. *Materials Cloud Archive* https://doi.org/10.24435/materialscloud:cv-zd (2021).
43. Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585**, 357–362 (2020).
44. Hirshfeld, F. L. Bonded-atom fragments for describing molecular charge densities. *Theoret. Chim. Acta* **44**, 129–138 (1977).
45. Baerlocher, C., Hepp, A. & Meier, W. Dls-76, a fortran program for the simulation of crystal structures by geometric refinement. *Institut fur Kristallographie und Petrographie, ETH, Zurich, Switzerland* (1978).
46. Pettifer, R., Dupree, R., Farnan, I. & Sternberg, U. NMR determinations of Si–O–Si bond angle distributions in silica. *Journal of Non-Crystalline Solids* **106**, 408–412 (1988).
47. Mauri, F., Pasquarello, A., Pfrommer, B. G., Yoon, Y.-G. & Louie, S. G. Si-O-Si bond-angle distribution in vitreous silica from first-principles 29 Si NMR analysis. *Physical Review B* **62**, R4786 (2000).
48. Wragg, D. S., Morris, R. E. & Burton, A. W. Pure silica zeolite-type frameworks: A structural analysis. *Chemistry of Materials* **20**, 1561–1570 (2008).
49. Ramdas, S. & Klinowski, J. A simple correlation between isotropic 29 si-nmr chemical shifts and t–o–t angles in zeolite frameworks. *Nature* **308**, 521–523 (1984).
50. Antao, S. M. Quartz: structural and thermodynamic analyses across the $\alpha \leftrightarrow \beta$ transition with origin of negative thermal expansion (NTE) in β quartz and calcite. *Acta Crystallographica Section B Structural Science, Crystal Engineering and Materials* **72**, 249–262 (2016).
51. OKeeffe, M. & Hyde, B. G. On Si–O –Si configurations in silicates. *Acta Crystallographica Section B* **34**, 27–32 (1978).
52. Bokeh Development Team. *Bokeh: Python library for interactive visualization*. https://bokeh.pydata.org/en/latest/ (2021).

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 814143. T.V. acknowledges funding of the research board of Ghent University. The computational resources (Stevin Supercomputer Infrastructure) and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by Ghent University, FWO and the Flemish Government–department EWI.

Author contributions

L.K. designed and performed the study. Both authors wrote the manuscript. T.V. oversaw the project.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to L.K. or T.V.

Reprints and permissions information is available at www.nature.com/reprints.

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

6

7 Conclusion & Outlook

Throughout this doctorate we designed and implemented ParAMS – a tool for the parametrization of chemical potential energy surface models. At its core ParAMS mitigates many technical issues that arise from conventional fitting workflows by providing an easy interface and model-independent data formats. This avoids convoluted and irreproducible parametrization scripts and allows researches to spend more time on the scientific side of the problem. Moreover, the ability to easily fit any computable physicochemical property to multiple empirical models is a novelty and a significant advantage of the package. For example, in a currently ongoing collaboration with the RWTH Aachen we have seen improved parametrization outcomes when replacing single point calculations of off-equilibrium structures with a relaxed PES scan – something that was not easily possible before.

An introduction to ParAMS alongside two first applications in the scientific literature was presented in Paper I (Section 5.1). In our second paper (Section 6.1) we presented a more complete workflow for a parametrization of the GFN1-xTB Hamiltonian [45], followed by a reference data set of ten thousand organosilicon compounds (Section 6.2). Both publications demonstrate how our software enables an easy entry into the field of model optimization, with a minimal, yet effective and reproducible user input.

As empirical and semi-empirical methods continue to become faster and more accurate, we expect ParAMS to maintain its relevance as a scientific tool for optimization. This is ensured by a number of novel design choices and functionality for this kind of software: (1) An intuitive and highly flexible syntax for the definition and storage of training and reference data. The syntax is model-independent and allows for an unprecedented diversity with regards to the construction of training data. Stored data is human-readable and clearly structured to help with reproducibility. (2) Coded in Python with a focus on user-friendliness, the package has a low entry barrier. At the same time, experienced users will be able to easily implement more complex workflows. Continuous integration ensures the package's stability at all times. (3) ParAMS is a modular software with well-defined interfaces. Not only does this make workflows highly re-usable, but also allows easy coupling of novel models and optimization algorithms into ParAMS. In conclusion, we

believe that ParAMS will become a valuable tool in a variety of future research projects.

Although ParAMS enables an easy entry into the field of parameter optimization, it is no replacement for sensible choices. For example, the fitting of thousands of parameters to only a few points is likely to be ineffective or lead to overfitting [70]. Also relevant outside of the world of chemistry, the problem of optimal training set design is scientifically intriguing but remains unaddressed. Detecting and removing uninformative entries from a training set would highly benefit the optimization process. A similar principle applies to the selection of parameters. Dimensionality reduction can be an effective approach for simplifying optimization problems. If a user has little to no information about the physical meaning of individual parameters, a method to determine a minimal, most sensitive subset can be a valuable tool. This is currently developed at our group by Michael Gustavo [71].

Methods that speed up the evaluation of the parameter search space are another interesting area of research. Even when optimizing fast empirical models, up to 10^6 parameter vector evaluations might be needed before a good solution is found. This results in wall times of several days to weeks, assuming each evaluation takes a few seconds. To address long computation times we considered the use of a surrogate model of the parameter space. Our goal was to estimate the loss of a candidate vector on the faster surrogate model and only calculate the real loss whenever the candidate was promising, consecutively rebuilding the surrogate after a number of evaluations. The project has not been fully developed, but first results were encouraging.

ParAMS can also be incorporated into larger workflows. Discussions with collaborators from the RWTH Aachen lead us to the design of an iterative setup. In it, a model with optimized parameters is used to drive a molecular dynamics simulation of a test system. After the simulation, the ChemTraYzer tool [72] is used to detect and extract chemical reactions. Reactions that were observed for the first time are added to the training set and a new optimization is started. The loop then repeats. Such and similar workflows are intended to provide good parameters in a mostly automated manner.

Bibliography

[1] Iowa State University. Location: EBBR. ASOS Network. https://mesonet.agron.iastate.edu/request/download.phtml?network=BE__ASOS.

[2] Muthukumar, M. Modeling polymer crystallization. *Interphases and mesophases in polymer crystallization III* 241–274 (2005).

[3] Gopferich, A. & Langer, R. Modeling of polymer erosion. *Macromolecules* **26**, 4105–4112 (1993).

[4] Gross, J. & Sadowski, G. Modeling polymer systems using the perturbed-chain statistical associating fluid theory equation of state. *Industrial & engineering chemistry research* **41**, 1084–1093 (2002).

[5] Kathrotia, T., Oßwald, P., Zinsmeister, J., Methling, T. & Köhler, M. Combustion kinetics of alternative jet fuels, part-iii: Fuel modeling and surrogate strategy. *Fuel* **302**, 120737 (2021).

[6] vom Lehn, F., Cai, L., Copa Cáceres, B. & Pitsch, H. Exploring the fuel structure dependence of laminar burning velocity: A machine learning based group contribution approach. *Combustion and Flame* **232**, 111525, https://doi.org/10.1016/j.combustflame.2021.111525 (2021).

[7] Bajorath, J. Integration of virtual and high-throughput screening. *Nature Reviews Drug Discovery* **1**, 882–894 (2002).

[8] Shoichet, B. K. Virtual screening of chemical libraries. *Nature* **432**, 862–865 (2004).

[9] Andricopulo, A. D., Guido, R. V. & Oliva, G. Virtual screening and its integration with modern drug design technologies. *Current medicinal chemistry* **15**, 37–46 (2008).

[10] Rao, R., Vrudhula, S. & Rakhmatov, D. N. Battery modeling for energy aware system design. *Computer* **36**, 77–87 (2003).

[11] Wang, Y. *et al.* A comprehensive review of battery modeling and state estimation approaches for advanced battery management systems. *Renewable and Sustainable Energy Reviews* **131**, 110015, https://doi.org/10.1016/j.rser.2020.110015 (2020).

[12] Jones, J. E. On the determination of molecular fields.—i. from the variation of the viscosity of a gas with temperature. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* **106**, 441–462 (1924).

[13] Jones, J. E. On the determination of molecular fields.—ii. from the equation of state of a gas. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* **106**, 463–477 (1924).

[14] Van Duin, A. C., Dasgupta, S., Lorant, F. & Goddard, W. A. Reaxff: a reactive force field for hydrocarbons. *The Journal of Physical Chemistry A* **105**, 9396–9409 (2001).

[15] Senftle, T. P. *et al.* The reaxff reactive force-field: development, applications and future directions. *npj Computational Materials* **2** (2016).

[16] Behler, J. & Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical Review Letters* **98**, 10.1103/physrevlett.98.146401 (2007).

[17] Smith, J. S. *et al.* Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning. *Nature Communications* **10**, 10.1038/s41467-019-10827-4 (2019).

[18] Devereux, C. *et al.* Extending the applicability of the ANI deep learning molecular potential to sulfur and halogens. *Journal of Chemical Theory and Computation* **16**, 4192–4202, 10.1021/acs.jctc.0c00121 (2020).

[19] Hornik, K., Stinchcombe, M. & White, H. Multilayer feedforward networks are universal approximators. *Neural networks* **2**, 359–366 (1989).

[20] Behler, J. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *The Journal of chemical physics* **134**, 074106 (2011).

[21] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M. & Monfardini, G. The graph neural network model. *IEEE Transactions on Neural Networks* **20**, 61–80, 10.1109/TNN.2008.2005605 (2009).

[22] Duvenaud, D. K. *et al.* Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems* **28** (2015).

[23] Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural message passing for quantum chemistry. In *International conference on machine learning*, 1263–1272 (PMLR, 2017).

[24] Deringer, V. L. *et al.* Origins of structural and electronic transitions in disordered silicon. *Nature* **589**, 59–64, 10.1038/s41586-020-03072-z (2021).

[25] Lu, D. *et al.* 86 PFLOPS deep potential molecular dynamics simulation of 100 million atoms with ab initio accuracy. *Computer Physics Communications* **259**, 107624, 10.1016/j.cpc.2020.107624 (2021).

[26] Shaw, D. E. *et al.* Millisecond-scale molecular dynamics simulations on anton. In *Proceedings of the conference on high performance computing networking, storage and analysis*, 1–11 (2009).

[27] Piana, S., Lindorff-Larsen, K. & Shaw, D. E. Atomic-level description of ubiquitin folding. *Proceedings of the National Academy of Sciences* **110**, 5915–5920 (2013).

[28] Wu, G., Robertson, D. H., Brooks, C. L. & Vieth, M. Detailed analysis of grid-based molecular docking: A case study of CDOCKER?a CHARMm-based MD docking algorithm. *Journal of Computational Chemistry* **24**, 1549–1562, 10.1002/jcc.10306 (2003).

[29] Hsin, K.-Y., Ghosh, S. & Kitano, H. Combining machine learning systems and multiple docking simulation packages to improve docking prediction reliability for network pharmacology. *PLoS ONE* **8**, e83922, 10.1371/journal.pone.0083922 (2013).

[30] Deringer, V. L., Caro, M. A. & Csányi, G. A general-purpose machine-learning force field for bulk and nanostructured phosphorus. *Nature communications* **11**, 1–11 (2020).

[31] Schrödinger, E. An undulatory theory of the mechanics of atoms and molecules. *Physical Review* **28**, 1049–1070, 10.1103/physrev.28.1049 (1926).

[32] Born, M. & Oppenheimer, R. Zur quantentheorie der molekeln. *Annalen der physik* **389**, 457–484 (1927).

[33] Hohenberg, P. & Kohn, W. Inhomogeneous electron gas. *Physical review* **136**, B864 (1964).

[34] Kohn, W. & Sham, L. J. Self-consistent equations including exchange and correlation effects. *Physical review* **140**, A1133 (1965).

[35] Kresse, G. & Furthmüller, J. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Computational Materials Science* **6**, 15–50, 10.1016/0927-0256(96)00008-0 (1996).

[36] Cramer, C. *Essentials of computational chemistry : theories and models* (Wiley, Chichester, West Sussex, England Hoboken, NJ, 2004).

[37] Elstner, M. & Seifert, G. Density functional tight binding. *Philos. T. R. Soc. A* **372**, 20120483, 10.1098/rsta.2012.0483 (2014).

[38] Elstner, M. *et al.* Self-consistent-charge density-functional tight-binding method for simulations of complex materials properties. *Phys. Rev. B* **58**, 7260–7268, 10.1103/PhysRevB.58.7260 (1998).

[39] Hourahine, B. *et al.* DFTB+, a software package for efficient approximate density functional theory based atomistic simulations. *J. Chem. Phys.* **152**, 124101, 10.1063/1.5143190 (2020).

[40] Szabo, A. & Neil, O. *Modern quantum chemistry : introduction to advanced electronic structure theory* (Dover Publications, Mineola, N.Y, 1996).

[41] Helgaker, T., Jørgensen, P. & Olsen, J. *Molecular Electronic-Structure Theory* (John Wiley & Sons, Ltd, 2000).

[42] Pople, J. A. Quantum chemical models (nobel lecture). *Angewandte Chemie International Edition* **38**, 1894–1902 (1999).

[43] Bohacek, R. S., McMartin, C. & Guida, W. C. The art and practice of structure-based drug design: a molecular modeling perspective. *Medicinal research reviews* **16**, 3–50 (1996).

[44] Mata, R. A. & Suhm, M. A. Benchmarking quantum chemical methods: Are we heading in the right direction? *Angewandte Chemie International Edition* **56**, 11011–11018, https://doi.org/10.1002/anie.201611308 (2017).

[45] 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). *Journal of Chemical Theory and Computation* **13**, 1989–2009, 10.1021/acs.jctc.7b00118 (2017).

[46] Wolfe, P. Convergence conditions for ascent methods. *SIAM review* **11**, 226–235 (1969).

[47] Wolfe, P. Convergence conditions for ascent methods. ii: Some corrections. *SIAM review* **13**, 185–188 (1971).

[48] Barzilai, J. & Borwein, J. M. Two-point step size gradient methods. *IMA journal of numerical analysis* **8**, 141–148 (1988).

[49] Nocedal, J. *Numerical optimization* (Springer, New York, 2006).

[50] Conn, A. R. & N., V. L. *Introduction to derivative-free optimization* (Society for Industrial and Applied Mathematics/Mathematical Programming Society, Philadelphia, 2009).

[51] Mayne, C. G., Saam, J., Schulten, K., Tajkhorshid, E. & Gumbart, J. C. Rapid parameterization of small molecules using the force field toolkit. *Journal of Computational Chemistry* **34**, 2757–2770, 10.1002/jcc.23422 (2013).

[52] Cosseddu, S. & Infante, I. Force field parametrization of colloidal CdSe nanocrystals using an adaptive rate monte carlo optimization algorithm. *Journal of Chemical Theory and Computation* **13**, 297–308, 10.1021/acs.jctc.6b01089 (2016).

[53] Shchygol, G., Yakovlev, A., Trnka, T., van Duin, A. C. T. & Verstraelen, T. ReaxFF parameter optimization with monte-carlo and evolutionary algorithms: Guidelines and insights. *Journal of Chemical Theory and Computation* **15**, 6799–6812, 10.1021/acs.jctc.9b00769 (2019).

[54] Dieterich, J. M. & Hartke, B. OGOLEM: Global cluster structure optimisation for arbitrary mixtures of flexible molecules. a multiscaling, object-oriented approach. *Molecular Physics* **108**, 279–291, 10.1080/00268970903446756 (2010).

[55] Hansen, N. & Ostermeier, A. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of IEEE international conference on evolutionary computation*, 312–317 (IEEE, 1996).

[56] Hansen, N. & Ostermeier, A. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation* **9**, 159–195 (2001).

[57] Chen, G., Li, S. & Wang, X. Source mask optimization using the covariance matrix adaptation evolution strategy. *Optics Express* **28**, 33371–33389 (2020).

[58] Bouzarkouna, Z., Ding, D. Y. & Auger, A. Well placement optimization with the covariance matrix adaptation evolution strategy and meta-models. *Computational Geosciences* **16**, 75–92 (2012).

[59] Akbarzadeh, V., Ko, A. H.-R., Gagné, C. & Parizeau, M. Topography-aware sensor deployment optimization with CMA-ES. In *Parallel Problem Solving from Nature, PPSN XI*, 141–150, 10.1007/978-3-642-15871-1_15 (Springer Berlin Heidelberg, 2010).

[60] Fujii, G., Akimoto, Y. & Takahashi, M. Exploring optimal topology of thermal cloaks by CMA-ES. *Applied Physics Letters* **112**, 061108, 10.1063/1.5016090 (2018).

[61] Hansen, N. The cma evolution strategy: A tutorial (2016). 1604.00772.

[62] Spicher, S. & Grimme, S. Robust atomistic modeling of materials, organometallic, and biochemical systems. *Angewandte Chemie International Edition* **59**, 15665–15673 (2020).

[63] Ponder, J. W. & Case, D. A. Force fields for protein simulations. *Advances in protein chemistry* **66**, 27–85 (2003).

[64] Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A. & Müller, K.-R. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics* **148**, 241722 (2018).

[65] Software for Chemistry & Materials (SCM). Amsterdam modeling suite 2021.4. https://scm.com (2019).

[66] Kim, S. *et al.* PubChem 2019 update: Improved access to chemical data. *Nucleic Acids Res.* **47**, D1102–D1109, 10.1093/nar/gky1033 (2018).

[67] Kim, S. *et al.* PubChem in 2021: new data content and improved web interfaces. *Nucleic Acids Res.* **49**, D1388–D1395, 10.1093/nar/gkaa971 (2020). https://academic.oup.com/nar/article-pdf/49/D1/D1388/35363961/gkaa971.pdf.

[68] Baerlocher, C. & McCusker, L. Database of zeolite structures. http://www.iza-structure.org/databases/.

[69] Raaijmakers, S., Pols, M., Vicent-Luna, J. M. & Tao, S. A reparameterized density functional tight-binding method for engineering phase-stable $CsPbX_3$ perovskites (2021). 2110.15827.

[70] Bishop, C. *Pattern recognition and machine learning* (Springer, New York, 2006).

[71] Freitas Gustavo, M. & Verstraelen, T. Reparameterization of computational chemistry force fields using glompo (globally managed parallel optimization). In *Learning and Intelligent Optimization*, 150–156 (Springer International Publishing, Cham, 2021).

[72] Döntgen, M. *et al.* Automated discovery of reaction pathways, rate constants, and transition states using reactive molecular dynamics simulations. *Journal of Chemical Theory and Computation* **11**, 2517–2524, 10.1021/acs.jctc.5b00201 (2015).

A Supporting Information

Supporting Information for 'ParAMS: Parameter Optimization for Atomistic and Molecular Simulations'

Leonid Komissarov,[†,‡] Robert Rüger,[‡] Matti Hellström,[‡] and Toon Verstraelen[*,†]

†*Center for Molecular Modeling (CMM), Ghent University, Technologiepark-Zwijnaarde 46, B-9052, Ghent, Belgium*

‡*Software for Chemistry & Materials (SCM) B.V., De Boelelaan 1083, 1081 HV Amsterdam, The Netherlands*

E-mail: toon.verstraelen@ugent.be

S1 The Parameterization Problem

This section provides a mathematical framework for the parameterization problem. We assume that the training data can be defined as a set of physico-chemical properties for a number of isolated or periodic systems. Examples for relevant properties are energy differences, nuclear gradients or system geometries. In the context of ParAMS, we define an arbitrary property P that can be expressed as the output of a computational job. When working with multiple jobs as part of a training set, a job function can be defined as

$$J(R_j, S_j, M) = \left(R'_j, P^n_j\right) \quad \forall\, j \in \{1 \ldots N_{\text{job}}\},\ n \in \{1 \ldots N_{\text{prop}}(j)\}, \tag{1}$$

calculating the output geometry R'_j and all properties P^n_j of a job j. The input for every job in J consists of the input geometry R_j, the job settings S_j (*e.g.* geometry optimization and

frequencies) and the computational model M. Note that a parametric model is additionally a function of the parameter vector $\boldsymbol{x}$, in which case the outputs of the above equation can be denoted with the hat operator (*i.e.* $\hat{R}'_j, \hat{P}^n_j$), as to distinguish between reference properties and properties predicted by the parametric model. Training set entries can be constructed, for example, from a linear combination of multiple properties

$$y_i = \sum_{k=1}^{N_{\mathrm{lc}}(i)} c_{i,k} P^{n(i,k)}_{j(i,k)} \quad \forall\, i \in \{1 \dots N_{\mathrm{data}}\}, \tag{2}$$

where $c_{i,k}$ is the coefficient for term k of training set entry i and $N_{\mathrm{lc}}(i)$ is the total number of terms per entry. Non-linear combinations of properties to construct y_i are also possible. Such a formulation offers a high degree of flexibility for the construction of a training set. One example is the combination of multiple system energies into one reaction energy. It should be noted that a training set entry, as defined in Eq. 2, does not have to originate from the results of computational jobs. The reference value can instead be provided directly, making it easy to work with experimental or external data.

While training set entries $\boldsymbol{y}$ have to be defined only once, their predicted counterpart $\hat{\boldsymbol{y}}$ has to be re-calculated every time the model parameters change. For this purpose, we introduce a Data Set function

$$DS(\boldsymbol{x}|\boldsymbol{y}) = \boldsymbol{y} - \hat{\boldsymbol{y}}, \tag{3}$$

which extracts all properties needed for the calculation of $\hat{\boldsymbol{y}}$ based on a parameter set $\boldsymbol{x}$ and returns the respective vector of residuals. A metric in the form of a loss function $L\Big((\boldsymbol{y}-\hat{\boldsymbol{y}})\boldsymbol{w}\Big)$ is then applied to the residuals for a qualitative measure of how close reference and predicted values are. The additional weights vector vector $\boldsymbol{w}$ can be used to balance possibly different orders of magnitude in the data set or make certain entries more relevant for the fitting process than others.

Finally, the optimization algorithm can be defined as a function that minimizes L with

respect to the parameters

$$O(\boldsymbol{x}_0, L) = \arg\min_{\boldsymbol{x}} L = \boldsymbol{x}^*, \tag{4}$$

finding an optimal solution $\boldsymbol{x}^*$ from an initial point $\boldsymbol{x}_0$.

S2 Additional Display Items

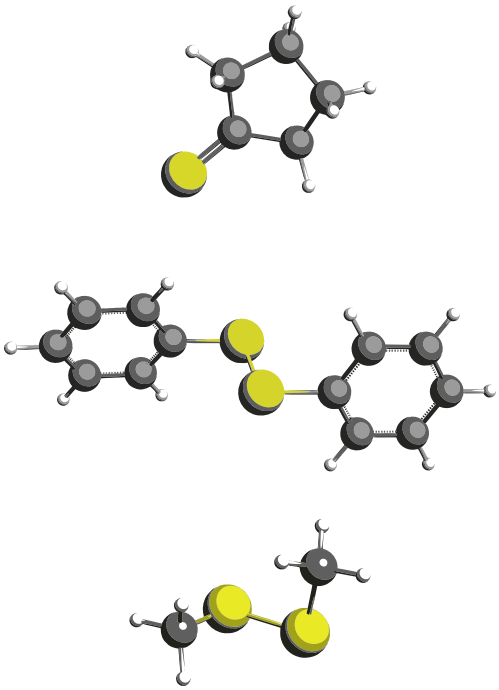


Figure S1: From top to bottom: Example structures of cyclopentathione, diphenyl disulfide and dimethyl disulfide, containing S (yellow), C (black), and H (white), included in the data provided by Müller and Hartke.[1] The fitted properties include bond distances, angles, relative energies and atomic forces.

Table S1: Composition of the reference data published by Müller and Hartke,[1] split by the computational tasks Single Point (SP) and Geometry Optimization (GO). For each of the two sets, the upper part describes the chemical systems, while the lower breaks down the individual entries in the training and validation sets. Note that some entries might be a function of multiple chemical systems, meaning that the sum of SP+GO is not necessarily equal to the total number of entries for that row (*cf.* Sec. 3.1 in the main text).

Training Set	**SP**	**GO**	**Total**
Number of systems	222	9	231
Mean system size (atoms)	6.6	11.4	6.8
Std. dev. (atoms)	2.9	7.7	3.3
Total number of entries	4620	317	4875
Energies	219	62	219
Forces	4401	0	4401
Atomic distances	0	94	94
Angles	0	85	85
Dihedrals	0	76	76
Validation Set			
Number of systems	200	24	224
Mean system size (atoms)	24.0	12.7	22.8
Std. dev. (atoms)	0.0	5.9	4.0
Total number of entries	199	771	970
Energies	199	0	199
Forces			0
Atomic distances	0	281	281
Angles	0	257	257
Dihedrals	0	233	233

Table S2: Summary of relevant ParAMS settings used for the re-parameterization of Mue2016.

Setting	**Value**
Number of optimizations	9
Number of parameters to optimize	35
Lower / upper parameter bounds	$\boldsymbol{x}_0 \pm 0.2\lvert\boldsymbol{x}_0\rvert$
Optimization timeout	24 hours
CMA-ES population size	36
CMA-ES sigma	0.3
Loss function	sum of squared errors
Early stopping patience	6000 evaluations
Constraints	$r_0^\sigma \geq r_0^\pi$ and $r_0^\pi \geq r_0^{\pi\pi}$

References

(1) Müller, J.; Hartke, B. ReaxFF Reactive Force Field for Disulfide Mechanochemistry, Fitted to Multireference ab Initio Data. *J. Chem. Theory Comput.* **2016**, *12*, 3913–3925.

Supporting Information

Improving the Silicon Interactions of GFN-xTB

Leonid Komissarov and Toon Verstraelen*

Center for Molecular Modeling (CMM), Ghent University, Technologiepark-Zwijnaarde 46, B-9052, Ghent, Belgium

E-mail: toon.verstraelen@ugent.be

S1. Extra display items

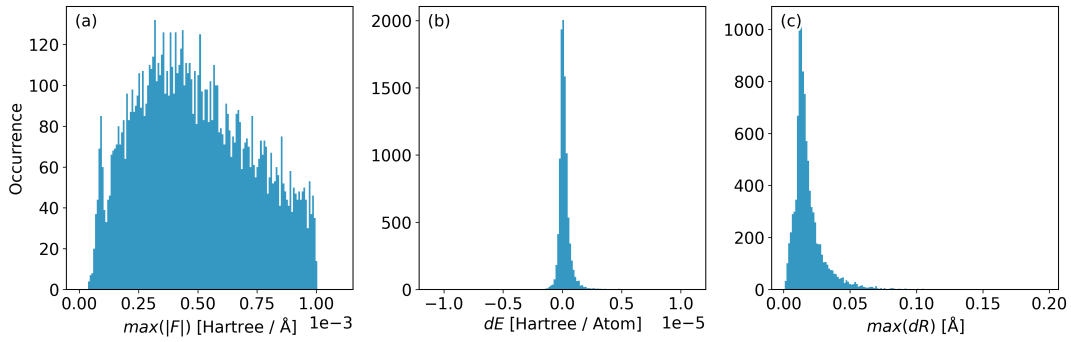


Figure S1: Distribution of convergence criteria at the last optimization step for all calculated systems in the reference data set. Showing (a) the highest absolute component of all nuclear gradients, (b) change in system energy and (c) highest relative atomic displacement.

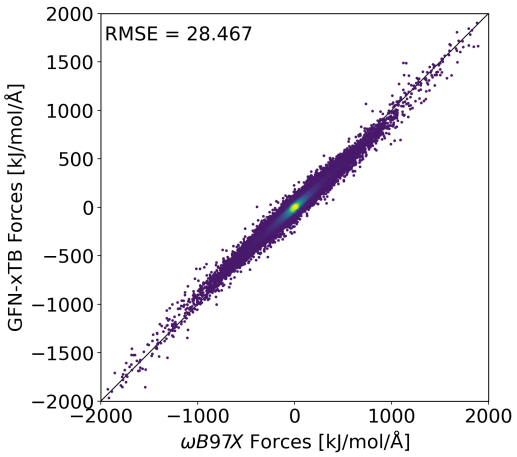


Figure S2: Correlation plot of atomic forces, as calculated with the test set generated from ANI-1x[1] data ($\omega WB97X$[2] reference, x-axis). As no silicon is present in this set, both, the GFN1-xTB and GFN1-xTB-Si parametrizations predict the same forces (y-axis).

Table S1: Format specification for the reference data repository. Each individual geometry optimization trajectory is stored in a NumPy .npz file with available keys listed below. Variables N and R denote the number of geometry optimization steps and the system size respectively.

Data	**Unit**	**Key**	**Array Shape**
Atomic Numbers	-	`numbers`	$(R,)$
Atomic Coordinates	Å	`xyz`	$(N, R, 3)$
Energy	hartree	`energy`	$(N,)$
Nuclear Gradients	hartree/bohr	`gradients`	$(N, R, , 3)$

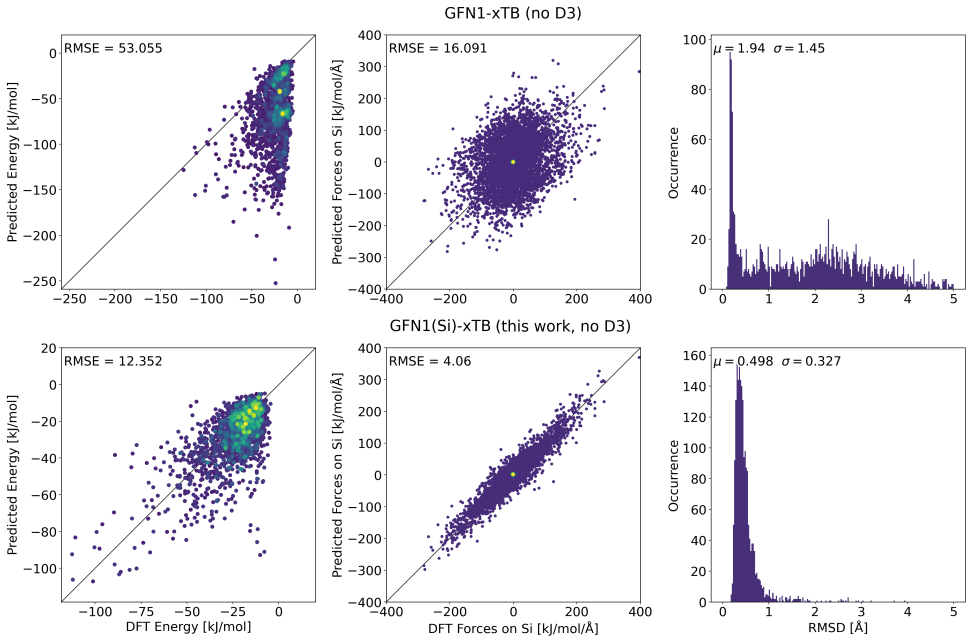


Figure S3: Validation set performance of GFN1-xTB[3] (top) and this GFN1(Si)-xTB (this work) (bottom) with D3-corrections disabled, showing comparable results to Fig. 2 in the main manuscript. Columns, from left to right depict energy differences, force components on the Si atoms, and RMSD of atomic positions (as described in the Methods Section). X and Y values in the first two columns are reference properties and their DFTB predictions respectively. Areas of lower point densities are depicted in dark blue; higher densities in bright green. Root-mean-square error (RMSE) printed in the same units as the axes. Histograms in the right column show the RMSD between geometry-optimized reference and DFTB structures. Mean μ and standard deviation σ printed in ångström.

References

(1) Smith, J. S.; Zubatyuk, R.; Nebgen, B.; Lubbers, N.; Barros, K.; Roitberg, A. E.; Isayev, O.; Tretiak, S. The ANI-1ccx and ANI-1x Data Sets, Coupled-Cluster and Density Functional Theory Properties for Molecules. *Sci. Data* **2020**, *7*, 1.

(2) Chai, J.-D.; Head-Gordon, M. Long-Range Corrected Hybrid Density Functionals with Damped Atom–Atom Dispersion Corrections. *Phys. Chem. Chem. Phys.* **2008**, *10*, 6615–6620.

(3) 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.* **2017**, *13*, 1989–2009.

B ParAMS Documentation

Many hours of this doctorate were spent documenting and creating tutorials for the ParAMS package. At the time of writing, the complete documentation contains over 250 A4 pages and is still updated daily. For this reason, we would like to refer the reader to `scm.com/doc/params` for the most up-to-date version. We would like to thank Dr. Matti Hellström, who has contributed a considerable part to the documentation, especially the recent Tutorials Section.

Members of the examination board can also access the pdf version of the documentation, submitted as a separate appendix.

C Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 814143.

The computational resources (Stevin Supercomputer Infrastructure) and services used in this work were partially provided by the VSC (Flemish Supercomputer Center), funded by Ghent University, FWO and the Flemish Government – department EWI.