A Kernel-based Learning Method for an Efficient Approximation of the High-Dimensional Born-Oppenheimer Potential Energy Hypersurface

Sonja Mathias

Born March 21, 1990 in Berlin

November 22, 2015

Master's Thesis Mathematics

Revised Version

Advisor: Prof. Dr. Griebel
Institute for Numerical Simulation

Mathematisch-Naturwissenschaftliche Fakultät der
Rheinischen Friedrich-Wilhelms-Universität Bonn

Declaration of Authorship

I hereby certify that this **revised** thesis has been composed by me and is based on my own work, unless stated otherwise. No other person's work has been used without due acknowledgement in this thesis. All references and verbatim extracts have been quoted, and all sources of information, including graphs and data sets, have been specifically acknowledged.

Uppsala, November 22, 2015

Abstract

In this thesis we describe and evaluate a kernel-based learning method for an efficient approximation of the high-dimensional Born-Oppenheimer potential energy hypersurface. To this end, the Coulomb matrix introduced by Rupp *et al.* in [45] is adjusted to represent local atomic environments and is used as input to the localised Gaussian process regression proposed by Bartók *et al.* in [4].

We show that this combination offers a promising approach for the generation of accurate and widely applicable potentials, by evaluating on small organic molecules as well as on silicon supercells. On the latter we obtain root mean squared errors in the total energy of the order of 1 meV per atom on small reference data sets of less than 500 configurations. We achieve mean absolute errors in atomisation energies on the biomolecular data set coming close to the desired chemical accuracy of 1 kcal/mol using only a fraction of the computational time needed for electronic structure calculations.

Additionally, we document the versatility of the generated potentials by demonstrating their transferability from small to larger molecules and their successful application to both finite and periodic chemical configurations. We test the significance of the localisation ansatz by systematically enlarging the atomic neighbourhood considered as input to the regression. Furthermore, we demonstrate their usefulness for molecular dynamics simulations by providing forces in addition to the energy.

Zusammenfassung

Das Ziel dieser Masterarbeit besteht in der Beschreibung und Evaluation eines kern-basierten Lernverfahrens zur effizienten Approximation der hochdimensionalen Born-Oppenheimer Energiehyperfläche. Dazu werden die von Rupp *et al.* in [45] eingeführten Coulomb Matrizen auf die Repräsentation von lokalen Atomumgebungen erweitert und als Eingabedaten für die lokalisierte Gaussprozessregression benutzt, die von Bartók *et al.* in [4] vorgestellt wurde.

Wir zeigen, dass dies ein vielversprechendes Verfahren darstellt, um genaue und vielseitig anwendbare Potentiale zu erzeugen, indem wir sowohl auf biomolekularen als auch auf Siliziumkristalldatensätzen auswerten. Auf letzteren erhalten wir mittlere Quadratfehler in der totalen Energie in der Größenordnung von 1 meV pro Atom für weniger als 500 Referenzdaten. Auf den Biomolekülen berechnen wir mittlere Absolutfehler der Atomisierungsenergie, die bereits nah an der angestrebten chemischen Genauigkeit von 1 kcal/mol liegen. Diese Ergebnisse erreichen wir in einem Bruchteil der Zeit, die man für Elektronenstrukturberechnungen benötigen würde.

Des Weiteren dokumentieren wir die Vielseitigkeit der erzeugten Potentiale, indem wir ihre Übertragbarkeit von kleinen auf größere Moleküle, sowie ihre erfolgreiche Anwendung auf endliche und periodische chemische Konfigurationen demonstrieren. Wir prüfen die Gültigkeit des Lokalisierungsansatzes, indem wir systematisch das Volumen der betrachteten Atomumgebungen vergrößern. Außerdem ergänzen wir die Vorhersage von Energiewerten durch die zugehörigen Kräfte, was eine wichtige Erweiterung für den Einsatz im Rahmen von Molekulardynamiksimulationen darstellt.

Contents

1. Introduction

The advancement of the computing power of modern hardware has opened the door to a new scale of simulation in terms of what is possible. Although in engineering and physics, the design and testing of new products with the help of theoretical modelling has already largely preceded costly trial-and-error experimentation, applications with processes on smaller scales have yet to profit from extensive simulation, as the problem posed by a large number of degrees of freedom coupled with suitable time resolution remains computationally challenging. Nevertheless, fields with such demands are situated at the forefront of scientific research, ranging from the understanding of biochemical processes (e.g. protein docking, protein folding, membrane diffusion) to the investigation of possible properties of future materials and the design of new drugs. Many of the important properties occur at scales described by quantum mechanical processes, which is why in chemistry, quantum physics and the material sciences, the efficient modelling of n-body systems and their dynamics has become a significant part of the scientific progress. The importance of this is underlined by the Nobel prize in Chemistry of 2013, which was awarded to M. Karplus, M. Levitt and A. Warshel for their "development of multiscale models for complex chemical systems" [35].

In the quantum mechanical theory, the state of such a system is described by the Schroedinger equation (SE). Unfortunately, it is not analytically solvable for systems with more than few atoms. In order to treat application-dictated problems with hundreds or thousands of particles, approximations must be made. In the last 50 years, extensive research has been conducted and a number of different approaches have been devised. These range from slow but accurate *ab initio* methods based on quantum mechanics, through not-so-slow semi-empirical methods where experimental data is combined with quantum mechanical calculations, to fast molecular mechanics, which model molecules as a collection of balls held together with springs. Whereas *ab initio* and semi-empirical methods approximate the wavefunction, the theoretical solution to the SE from which the electronic distribution can be calculated, another approach called density functional theory (DFT) aims to derive an electron density directly. Of course, the method of choice depends on the accuracy required, as well as on time and resources that can be invested to solve the problem at hand.

In the field of molecular dynamics (MD) the most fundamental simplification applied to the SE is the Born-Oppenheimer-Approximation. It postulates that due to their huge difference in mass, the time scales of the dynamics of electrons and nucleus become decoupled; hence the electrons "see" the nuclei as static. The nuclei in turn can be modelled as point particles, moving on the electronic

potential energy surface (PES) according to the laws of Newtonian physics. As a further simplification only the ground state is studied. In this context, the main problem becomes the construction of the electronic potential. Solving the electronic SE or even evaluating DFT methods for every single timestep remains intractable, so over the years many different empirical potentials have been tailored for specific applications. Starting with simple pair potentials that only consider the interactions between two particles, e.g. the Coulomb potential, which models the interactions of point charges, or the Lennard-Jones potential used to model van der Waals interactions, current state-of-the-art potentials are many-body potentials taking into account also the interactions between three and more particles [37]. All of these potentials provide excellent results for their intended purposes; however, they are not transferable between applications.

Hence, the search for general, accurate and efficient potentials remains an active problem. Its resolution would have a huge impact on both material science and computational chemistry, and is therefore tackled from many different angles. Besides the experimental approach leading to the above mentioned empirical potentials, classical theory to solve inverse problems in the framework of reproducing kernel Hilbert spaces has been applied to interpolate the PES as a high-dimensional function (cf. [22] and the references therein), as have been more sophisticated techniques as the modified Shepard's interpolation [7].

In recent years, like many other fields, computational chemistry has profited from the introduction of Machine Learning (ML) techniques. Those have two primary objectives: pattern recognition and prediction, both of which have been successfully applied. The first is used for so-called "high-throughput" computational materials design. Large combinatorial databases are constructed comprising a wealth of different materials, too many to analyse by hand so instead data mining is used to methodically extract information and find desired materials [12]. The second purpose, prediction, is closely related to the field of interpolation. It is used to infer structure-property relations from historic data and then apply them to new systems. This approach is highly promising for the generation of potentials. Its advantages include that while the calculation of the training data using DFT or even *ab initio* methods and the training of the potential might be costly, evaluation can be expected to be comparatively cheap and fast. The task of solving the SE is completely circumvented. Nevertheless, the accuracy of such a potential is only limited by the accuracy of the underlying method and the number of training examples. Hence, they can be expected to come near to the accuracy of DFT methods for a fraction of their computational cost.

Furthermore, many ML predictive methods possess the means to evaluate their own quality. This can be used to construct feedback algorithms that automatically learn carefully chosen data points to improve their performance, ensuring a maximum benefit for a given number of costly DFT calculations. This is important not only to save computational resources, but also since the predictive power of a learned potential depends strongly on the new system not being qualitatively distinct from the training data. In theory, the validity of a potential

is independent of the knowledge of the underlying physics and can be generalised indefinitely as long as enough training data can be provided. In practice, of course, resources are limited, requiring individually-learned instances of the same potential for different structures.

The whole concept of constructing a transferable potential only works assuming a suitable descriptor of the structure underneath has been found capturing all relevant information. The choice of such a descriptor is highly non-trivial and has to fulfill several requirements concerning physical aspects like rotation and translation invariance, but also computational ones, e.g. it should not be more costly to calculate than the potential evaluation itself. Basing such a descriptor only on *ab initio* information of the system, i.e. including no structural or bonding information, carries the promise of generating potentials valid for all types of compounds and even chemical reactions. Beyond that, one needs to define a notion of chemical similarity or rather dissimilarity to allow the potential to relate the features of a new structure to those of the training data it was built with.

Hence, in addition to the choice of ML methods, such as Neural Networks (NN), Kernel Ridge Regression (KRR), or Bayesian inference (BI), there are many decisions to be made when designing a mathematical potential. The next section gives an overview over the different approaches used to construct potential energy hypersurfaces found in the literature. We refer also to the review article by Handley and Behler, [17], focusing on the application of condensed systems that summarises past and recent techniques.

Of course, the application of ML methods to material science is not limited to the prediction of energy values. There have been numerous examples of promising approaches, such as the prediction of crystal structures in bulk materials [13], [15], the approximation of density functionals [50], or the optimisation of transition states [39].

1.1. Using Machine Learning for PES Interpolation

The earliest efforts to introduce Machine Learning techniques for the construction of the high-dimensional PES were based on neural networks. Multilayer neural networks allow for the representation of highly complex functional relationships with great accuracy. As they have the downside of being very difficult to train, successful learning depends strongly on the initial configuration of the weights and on the description of the data. Nevertheless, they have been employed successfully since the mid-1990s (see [18] for a elaborate overview) and are still being investigated by several groups for the use for surface reactions, [28], or water clusters [34]. One of the most promising attempts with respect to the transferability of the potential is the work of Behler and Parinello [6].

A different approach to constructing an automatically generated interatomic potential from quantum mechanical data was proposed by Bartók *et al.* in

2010 [4]. They employed Gaussian process (GP) regression based on a decomposition into individual atomic energies to interpolate the PES, using a modified bispectrum of the atomic density as a descriptor of local atomic environments within a specific cutoff. Gaussian Process regression is a Bayesian inference method that combines prior knowledge with the likelihood of the observed data to obtain a posterior probability distribution used for prediction. Testing of this Gaussian Approximation Potential (GAP) was done on the bulk phases of semiconductors, such as carbon, silicon and germanium, leading to root mean squared errors in the energy of less than 1 meV/atom from the reference DFT calculations. The localisation ansatz present in the construction of the GAP limits its predictive power for processes where long-range interactions play an important role. Bartók *et al.* circumvent this by explicitly adding Coulomb and dispersion terms to the total energy expression. Similarly, the GAP framework can be applied to learn only correction terms to the total energy in order to improve accuracy of DFT methods [2]. As an improvement to their framework, Bartók *et al.* studied the problem of deriving continuous invariant representations of atomic neighborhoods more carefully and generalised the concept to an improved similarity measure called Smooth Overlap of Atomic Positions (SOAP) [3], in which the descriptor is implicitely embedded. This similarity measure was used to construct a GAP for tungsten viable for a large range of properties [53].

Recently, Thompson *et al.* have submitted a paper [54], in which they propose a quantum-accurate potential called SNAP similar to the GAP. While also using the bispectrum, they perform a linear least squares fit against *ab initio* data to predict the energy values.

In 2012, Rupp *et al.* published an ML ansatz to predict molecular atomisation energies using KRR [45], a versatile method equivalent to Gaussian Process regression but lacking the Bayesian framework. Their goal was to provide a general learning scheme valid for ground-state energies of molecules distributed over a vaste region of the chemical compound space (CCS), rather than to accurately interpolate the PES for just one structure as done by those mentioned before. As global descriptor, they introduced the so-called Coulomb matrix based only on the nuclear charges and cartesian coordinates of a molecule. They were able to achieve a MAE of about 10 kcal/mol using cross validation over more than 7000 small organic molecules. They refined their descriptor to account for permutation invariance [32] and used it to train a multitask NN to predict several electronic properties simultaneously [33]. In [20], Hansen *et al.* conducted an extensive comparison between different ML methods including NN and KRR to predict atomisation energies using different variants of the Coulomb matrix as descriptor, leading to MAEs between 3 and 9 kcal/mol, already approaching the required chemical accuracy of 1 kcal/mol for *in silico* rational molecular design. They were able to improve the results to 1.5 kcal/mol with the Bag-of-Bonds model [19], which represents the molecule as a concatenated vector of Coulomb matrix-like entries for each pair, when using KRR with a Laplacian kernel.

Schuett *et al.* suggested possible expansions to the Coulomb matrix approach to periodic structures [48]. Due to the lacking predictive power of these Crystal Coulomb Matrices, they opted for a different ansatz using partial radial distribution functions considering the distribution of pair-wise distances between atom types, which gave promising results in combination with KRR.

Atomic radial distribution functions were also used as arguments of a Fourier series to build a first principles-like descriptor dubbed FGR satisfying all crucial properties such as uniqueness, invariance and differentiability [56]. Its predictive power, when used in KRR, was shown to be on par with that of the Coulomb matrix.

1.2. Combining the GAP with the Localised Coulomb Matrix

The goal of this thesis is to propose an interpolation ansatz combining the powerful concept of GAPs with the simple yet efficient Coulomb matrix. Due to the localisation ansatz inherent to the GAP, it is applicable to both periodic and non-periodic structures, implying the biggest generality across chemical compound space possible. Nevertheless, it can only benefit from this when using a powerful descriptor capturing all necessary information of an atomic neighbourhood. While the bispectrum and the SOAP distance measure proposed by Bartók *et al.* exhibit all crucial features, they are overly complicated in construction, in contrast to the Coulomb matrix proposed by Rupp *et al.* As has been shown, it has a strong predictive power inspite of its simple definition.

However, the Coulomb matrix is a global descriptor of molecules. In order to apply it to atomic environments independently of the type of chemical compound, it has to be localised in a suitable way. To this end, we introduce different variants of a localised version and test their performance in the GAP framework both on the GDB-7 data set used by Rupp *et al.* [45], and on bulk configurations of silicon as done by Bartók *et al.* [4]. To the best of our knowledge, this is the first time that a matrix based descriptor of atomic environments is successfully used for predicting the energies of both organic and crystalline structures.

We are able to show that the resulting potential, which we call *Localised Coulomb matrix based Gaussian Approximation Potential (LC-GAP)*, fulfills all the necessary requirements for being efficiently employed in large scale molecular dynamics simulations. Not only is it accurate, general and transferable, it is also of linear complexity in the number of particles when predicting the energy and forces for a given system. This is a substantial challenge for the computer simulation of large particle systems composed of thousands of atoms, which is an important problem in computational biochemistry and material science.

We obtain a predictive power in atomisation energy of organic molecules that surpasses the reference values stated by Hansen *et al.* in [20] not only by a factor

of two in absolute value, but also by a factor of ten in training set sizes. Moreover, the localisation ansatz allows for the accurate prediction of molecules featuring up to seven heavy atoms using a potential trained on molecules composed only of up to five. The LC-GAP is thus a promising means for the so-called "learning across compound space" that aims to attribute physical properties to any chemical stoichiometry.

The main contributions of this thesis can be summarised as follows:

- Summary of the relations between Gaussian process regression and regularisation in the context of inverse problems
- Detailed presentation of the underlying localised GP regression of the GAP framework proposed by Bartók-Pártay in [4] based on unobservable atomic energy contributions
- Design and implementation of a new descriptor of atomic neighbourhoods called localised Coulomb matrix, based on the global Coulomb matrix introduced by Rupp *et al.* in [45]
- Validation of the predictive power of the GAP when combined with the localised Coulomb matrix on two different data sets as to their performance concerning different parameters such as kernel, cut-off radius and degree of localisation

This thesis is structured as follows. In Chapter 2 we start by summarising the mathematical basis of the Gaussian process regression as a Bayesian inference method and relate it to the classical approach to multivariate function approximation in the context of inverse problems.

We then apply it to the interpolation of the potential energy hypersurface via an atomistic decomposition of the total energy in Chapter 3. The representation of the atomic energy as a linear combination of kernel functions is derived in detail. Additionally, we describe the extension to the prediction of gradients and their incorporation into the training data, as well as to the learning of energy differences arising as the corrections to empirical potentials.

Chapter 4 is concerned with the issue of numerically representing chemical environments. The physical and computational requirements of such a descriptor are analysed and current local descriptors used as input to Machine Learning methods are presented. We introduce a new descriptor of atomic environments called *localised Coulomb matrix* based on the Coulomb Matrix by Rupp *et al.* and discuss its properties and possible variants in detail.

The assessment and validation procedure of the proposed framework is described in Chapter 5. This includes description of the data sets and their preparation, implementation details of the localised Coulomb matrix and the Gaussian Approximation Potentials, as well as the analysis of their computational cost.

The presentation of the numerical results is done in Chapter 6. First, the biomolecular data sets derived from the GDB-7 data set are used to assess the

performance of the different variants of the localised Coulomb matrix as well as their localisation properties and the transferability of the resulting GAPs. Then, the ability of the localised Coulomb matrix to handle crystalline structures is tested using the silicon data set, and the prediction is extended to gradients.

Finally, conclusions are drawn and outlooks given in Chapter 7. We conclude with a discussion of potential future extensions to the LC-GAP.

2. Multivariate Function Approximation

Infering a smooth function from sparse data is an important task in many mathematical applications. As such, it has a long history of being treated in the context of inverse problems using regularisation theory [29].

A newer approach is that of statistical learning theory, a subfield of supervised Machine Learning. Here, it is one of the core problems known under the term *regression*, i.e. the prediction of a continuous function $f : \mathbb{R}^d \to \mathbb{R}$ based only on finitely many noisy data points $\mathcal{D} = (X, \mathbf{y}) = \{\mathbf{x}_n, y_n\}_{n=1,\dots,N}$ with

$$y_n = f(\mathbf{x}_n) + \varepsilon. \tag{2.1}$$

Numerous different methods have been proposed to tackle it, for an overview see [31].

Since the 1990s so-called kernel-based methods have become popular as they allow for a decoupling of learning and data representation in a modular fashion [47]. The kernel efficiently computes the similarity of the raw input by implicitely projecting them onto a possibly high-dimensional feature space. This is called the *kernel trick*. By using different kernels the framework becomes very flexible.

Interestingly, the classical and the kernel-based learning approach are closely connected. We analyse this relation using the example of Gaussian Process Regression, a popular Bayesian kernel-based learning method. We start by presenting the mathematical basis of the Bayesian approach to statistical learning theory in Section 2.1. Section 2.2 deals with multivariate function approximation in the context of inverse problems. The connection between the two approaches is then made clear in Section 2.3.

2.1. Bayesian Inference

For a reference for the Bayesian approach to multivariate function approximation in the Machine Learning context see [47].

Bayesian inference is an important technique in mathematical statistics. It produces a prediction as a combination of prior beliefs and information gained by sampling data by modelling both to be controlled via an underlying probability distribution.

It thereby provides an intuitive way to include any information one might have about the function to be reconstructed. It incorporates these assumptions into a *prior* distribution over the *latent*, i.e. the underlying function values. Here, the term *prior* indicates that this distribution is specified without having seen any instances of the data. Once the data is sampled, its *likelihood*, i.e. the probability of the observations given the true function f, is specified. The observations may differ from the latent function values by noise. The prior is then combined with the likelihood using Bayes' rule to obtain the *posterior* distribution which is used for prediction. The term *posterior* is used to emphasise that this distribution includes knowledge about the data. It quantifies how plausible functions appear after data has been seen.

In detail, inference is performed as follows:

1. Put a joint prior $p(f, f^\star)$ on latent training and test values $f, f^\star$,
2. Specify likelihood $p(y|f)$ of observations given the latent training values,
3. Obtain joint posterior using Bayes' formula

$$p(f, f^\star|y) = \frac{p(f, f^\star)p(y|f)}{p(y)}, \tag{2.2}$$

4. Produce desired posterior predictive distribution by marginalisation of the unwanted training set latent variables

$$p(f^\star|y) = \int p(f, f^\star|y)df = \frac{1}{p(y)} \int p(f, f^\star)p(y|f)df, \tag{2.3}$$

5. Use its mean as the prediction value for $f^\star$.

One of the main advantages of the Bayesian framework is its ability to assess the accuracy of its own prediction via the variance of the posterior predictive distribution. This can be used for so-called *learn-on-the-fly* algorithms that iteratively improve the accuracy of the approximation by suitably choosing the training data.

In practice, however, the biggest difficulty of Bayesian inference lies in the evaluation of the involved integrals. Depending on the distributions chosen as prior and likelihood, the resulting posterior may or may not be analytically tractable. For the case that evaluating the integral is computationally demanding, many approximation techniques such as the *maximum a posteriori* (MAP) approximation exist. Here the posterior is written as

$$p(f^\star|y) = \int p(f^\star|f)p(f|y)df, \tag{2.4}$$

and then the integral over $p(f|y)$ is replaced by its *mode*, i.e. the value for which $p(f|y)$ is maximal. This leads to the following approximation

$$p(f^\star|y) \approx p(f^\star|f_{MAP}), \text{ where } f_{MAP} = \operatorname{argmax}_f p(f|y). \tag{2.5}$$

Another possibility is numerical integration via Monte Carlo methods [43].

We will see in the following subsection that one of the main reasons for the popularity of the Gaussian Process regression is that the choice of normal distribution allows for all integrals to have closed analytical forms. Hence, we have no need for any approximations.

2.1.1. Gaussian Process Regression

Gaussian process regression models the training data as a Gaussian process (GP), meaning it assumes that the observed function values are jointly normally distributed. The huge advantage of choosing this prior over any other arbitrary distribution is the analytical and, as a result, computational tractability. An introductory reference to Gaussian process regression is the book by Rasmussen [42]. For the definition of the basic probability concepts used, refer to Appendix A.

We start by defining the notion of a Gaussian process. Informally, a Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution. We now state this more rigorously.

Definition 1 (Gaussian Process)**.**
Let I be an arbitray index set. A collection $(Y_t)_{t\in I}$ of random variables $Y_t : \Omega \to \mathbb{R}$ defined on a probability space $(\Omega, \mathcal{A}, P)$ is called a Gaussian process *if and only if the joint distribution of any finite subcollection $Y_{t_1}, ..., Y_{t_n}$, $n \in \mathbb{N}$, $t_1, ..., t_n \in I$ is a multivariate normal distribution.*

We note that the distribution of a Gaussian process $(Y_t)_{t\in I}$ is uniquely determined by the distributions of finite subcollections and hence by the expectation values

$$m(t) = \mathbb{E}[Y_t], \;\; t \in I, \tag{2.6}$$

and the covariances

$$c(s,t) = Cov[Y_s, Y_t] := \mathbb{E}[(Y_s - m(s))(Y_t - m(t))], \;\; s,t \in I. \tag{2.7}$$

We will consider only centered Gaussian processes, i.e. processes with $m(t) = 0$.

Modelling the training data as a Gaussian process simply means that we interprete the observations as random variables $(f(x))_{x\in I}$, with the index set I corresponding to the input space $\mathcal{X}$, i.e. $I = \mathcal{X} = \mathbb{R}^D$. For a given finite index set $x_1, ...x_N$ the vector of corresponding observations $f(x_1), ..., f(x_N)$ consequently has a multivariate Gaussian distribution (cf. Appendix B.1)

$$\mathbf{f} := (f(x_1), ..., f(x_N)) \sim \mathcal{N}(0, \mathbf{C}), \tag{2.8}$$

where $\mathbf{C}$ is the covariance matrix with the entries

$$\mathbf{C}_{ij} = \mathrm{Cov}(f(x_i), f(x_j)) =: \mathcal{k}(x_i, x_j). \tag{2.9}$$

Thus, the GP model allows us to express the covariance between two outputs as a function $\mathcal{k}$ of the corresponding inputs which evaluates their "similarity". Mathematically, another term for a function mapping two inputs into $\mathbb{R}$ is *kernel*. We will rigorously define the notion in the subsequent subsection.

In practice, one is often faced with the additional problem of not measuring the actual latent function values $f(x_i)$, but some observation y_i corrupted by noise ε_i. In the context of Bayesian inference this noise is not considered to be deterministic but random according to some distribution. The suitable choice of this distribution is by no means an easy question and depends heavily on the application. For simplicity, we will assume Gaussian white noise, i.e. that the noise is independent and identically normally distributed over the training data,

$$y_i = f(x_i) + \varepsilon, \text{ with } \varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon). \tag{2.10}$$

This choice has the main advantage that all posterior distributions obtained by the Bayesian inference remain normal, allowing for the computation of exact solutions.

Since the observed function values $\mathbf{y}$ are thus the sum of two Gaussian random variables, they in turn are normally distributed as

$$\mathbf{y} \sim \mathcal{N}(0, \mathbf{C} + \sigma_\varepsilon^2 I). \tag{2.11}$$

We now introduce a test input $x_\star$ for which we wish to predict the function value $f_\star := f(x_\star)$. Adding this value to our finite index set, we obtain the joint prior on the latent training and test values consistently as

$$(\mathbf{y}, f^\star) \sim \mathcal{N}\left(0, \begin{pmatrix} \mathbf{C} + \sigma_\varepsilon^2 I & \mathbf{k}_\star \\ \mathbf{k}_\star & k_{\star\star} \end{pmatrix}\right), \tag{2.12}$$

where $\mathbf{k}_\star = (\mathcal{k}(x_i, x_\star))_{i=1}^N$ denotes the covariance between the training and test values and $k_{\star\star} = \mathcal{k}(x_\star, x_\star)$ the variance of the test value itself.

The posterior predictive distribution is now obtained by conditioning $f_\star$ on the observed function values $\mathbf{y}$. Using the relations holding true for multivariate Gaussian distributions found in Appendix B.2, it can be calculated as

$$f^\star | \mathbf{y} \sim \mathcal{N}\left(\mathbf{k}_\star (\mathbf{C} + \sigma_\varepsilon^2 I)^{-1} \mathbf{y}, k_{\star\star} - \mathbf{k}\star (\mathbf{C} + \sigma_\varepsilon^2 I)^{-1} \mathbf{k}_\star\right). \tag{2.13}$$

Its mean value $\mathbf{k}_\star (\mathbf{C} + \sigma_\varepsilon^2 I)^{-1} \mathbf{y}$ is used as the prediction value for $f_\star$. We will identify both and henceforth write

$$f_\star = f(x_\star) = \mathbf{k}_\star (\mathbf{C} + \sigma_\varepsilon^2 I)^{-1} \mathbf{y}. \tag{2.14}$$

Looking closely at this expression, we observe that the reconstructed function is written as a linear combination of the kernel functions fixed by the data points,

$$f(x^\star) = \sum_{i=1}^{n} \alpha_i \mathcal{k}(x_i, x^\star), \text{ with } \alpha = (\mathbf{C} + \sigma_\varepsilon^2 I)^{-1}\mathbf{y}. \tag{2.15}$$

The learning of this algorithm actually consists in the calculation of the coefficients α via a matrix inversion of the corrupted covariance matrix. This means that the choice of the kernel function determines the space in which f is reconstructed. This issue will be addressed in more detail when considering multivariate function inference in the context of inverse problems in Section 2.2.

We will now present different covariance functions popular in the machine learning community.

Covariance Functions

The covariance function ensures the basic assumption in supervised learning, namely that close inputs lead to similar target values. It has to provide a measure of closeness for inputs on which the prediction can be based. Hence the choice of the covariance function is crucial for the predictive quality of the Gaussian process regression.

We will now provide a rigorous definition of the notion of a kernel and then use the two terms *covariance function* and *kernel* interchangeably. For a more detailed analysis of the subject we refer the reader to [47].

Definition 2.
Let $\mathcal{X}$ be a non-empty set. A function $\mathcal{k} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a (positive definite) kernel *if for all $M \in \mathbb{N}$ and all $x_1, ..., x_M \in \mathcal{X}$ the matrix*

$$\mathcal{K} = \{\mathcal{k}(x_i, x_j)\}_{i,j=1}^{M}$$

is symmetric and positive semidefinite.

One can easily show that the covariance function of a Gaussian process defines an admissible kernel.

As already noted at the beginning of this chapter under the term *kernel trick*, kernels can also be understood as mappings into an implicit feature space. In fact, every kernel can be represented as a dot product in some space $\mathcal{H}$ by virtue of

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}}, \tag{2.16}$$

where $\Phi : \mathcal{X} \to \mathcal{H}$ is called the feature map and defines the space $\mathcal{H}$. In Section 2.2.2 we will describe the construction of the feature space in more detail by choosing Φ as the map $x \mapsto k(., x)$.

There are many different types of kernels satisfying the above definition. In our application we will require *stationary* kernels, i.e. functions invariant to

translations in input space. Clearly, such kernels only depend on the difference $x - x'$.

A standard choice for such a kernel is the squared exponential, or Gaussian kernel

$$\mathcal{k}(x, y) = \sigma_f \exp\left(-\frac{||x - y||_2^2}{2l^2}\right), \tag{2.17}$$

where l is called the characteristic length scale and σ_f denotes its amplitude. Note that there is no dependence between the Gaussian prior on the latent training values and the choice of the Gaussian kernel as a covariance function for the resulting Gaussian process. The Gaussian prior assumption renders the model analytically tractable, while for the covariance function other choices are possible.

An alternative stationary kernel is e.g. the Laplacian kernel,

$$\mathcal{k}(x, y) = \exp\left(-\frac{|x - y|_1}{l}\right). \tag{2.18}$$

It has a similar form as the Gaussian kernel but uses the ℓ_1 norm in the exponent of the exponential map leading to the slight disadvantage of not being continuously differentiable.

Both the Laplacian and the Gaussian kernel can be interpreted as special cases of the more general Matérn class of kernels, named after the work of Matérn [30]. They are given by

$$\mathcal{k}(x, y) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}|x - y|}{l}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}|x - y|}{l}\right), \tag{2.19}$$

where ν and l are positive parameters and K_ν denotes the modified Bessel function. The Gaussian kernel is recovered in the limit $\nu \to \infty$ and the Laplacian kernel corresponds to $\nu = \frac{1}{2}$. For half-integer ν the functional form simplifies into a product of a polynomial and an exponential term. Examples are

- $\nu = \frac{3}{2} : \mathcal{k}(x, y) = (1 + \frac{\sqrt{3}}{l}|x - y|)\exp(-\frac{\sqrt{3}}{l}|x - y|)$,
- $\nu = \frac{5}{2} : \mathcal{k}(x, y) = (1 + \frac{\sqrt{5}}{l}|x - y| + \frac{5}{3l^2}|x - y|^2)\exp(-\frac{\sqrt{5}}{l}|x - y|)$.

So far all kernels mentioned are *isotropic*, meaning they are functions of $r = |x - x'|$ which weigh all input dimensions equally. Depending on the input, it can be reasonable to relax this assumption and assign different length scales to particular dimensions. The anisotropic Gaussian kernel for example is defined as

$$\mathcal{k}(x, y) = \exp\left(-\sum_{i=1}^{D}\frac{(x_i - y_i)^2}{2l_i^2}\right) \tag{2.20}$$

for a vector $\mathbf{l} = (l_i)_{i=1}^D$ of characteristic length scales.

Marginal Likelihood

Having chosen a kernel, the question remains how to chose its parameters. As these are parameters of the prior, they are often refered to as *hyperparameters* in order to distinguish them from possible parameters of the model itself. The Bayesian framework provides a rather intuitive way to select the best hyperparameters for a given set of training data.

Consider the likelihood of the observed function values $p(y)$. It is also called *marginal* likelihood as it can be obtained by marginalisation of the latent function values,

$$p(y) = \int p(y|f)p(f)df. \tag{2.21}$$

By our choice of priors and noise distribution it is normally distributed,

$$p(y) = \mathcal{N}(0, \mathbf{C} + \sigma_\varepsilon^2 I). \tag{2.22}$$

To be accurate, the observation values actually depend both on the input X and on the hyperparameters Θ of the model,

$$p(y|X, \Theta) = \mathcal{N}(0, \mathbf{C}(\Theta) + \sigma_\varepsilon^2 I) \tag{2.23}$$

In order to determine the best set of parameters, we can maximise the likelihood $p(\Theta|y, X)$ of the hyperparameters given the observed training data. Using Bayes' rule, it can be calculated as the normalised product of marginal likelihood and hyperparameter prior,

$$p(\Theta|y, X) = \frac{p(y|X, \Theta)p(\Theta)}{p(y)}. \tag{2.24}$$

If we assume that all hyperparameters are initially equally likely (without having seen the training data), i.e. if $p(\Theta) = const$, then the arguments of the maximum of the hyperparameter likelihood and the marginal likelihood coincide,

$$\operatorname{argmax}_\Theta p(\Theta|y, X) = \operatorname{argmax}_\Theta p(y|X, \Theta). \tag{2.25}$$

Hence, it suffices to maximise the marginal likelihood. This is facilitated by applying the logarithmic map - which is monotonic - as the products then transform into sums.

In practice, one minimises the negative *loglikelihood* of the density function of the multivariate normal distribution (c.f. B.1) given by

$$-\ln p(y|\Theta) = \frac{1}{2}\ln|\mathbf{C}(\Theta) + \sigma_\varepsilon^2 I| + \frac{1}{2} y \left(\mathbf{C}(\Theta) + \sigma_\varepsilon^2 I\right)^{-1} y + \frac{m}{2}\ln 2\pi \tag{2.26}$$

Its terms can be interpreted as follows. The first term depends on the determinant of the perturbed covariance matrix. Therefore, it penalises the complexity

of the chosen model. The second term is the data fit term and the third only a normalisation.

A necessary condition for an extremum are vanishing gradients. Therefore, the it is helpful to compute the derivatives of the loglikelihood with respect to the hyperparameters as

$$\begin{aligned}\frac{\partial \ln p(y|\Theta)}{\partial \theta_i} =& \frac{1}{2\left|\mathbf{C}(\Theta)+\sigma_\varepsilon^2 I\right|} \operatorname{tr}\left(\mathbf{C}(\Theta)+\sigma_\varepsilon^2 I\right) \frac{\partial\left(\mathbf{C}(\Theta)+\sigma_\varepsilon^2 I\right)}{\partial \theta_i} \\ &+ \frac{1}{2} y\left(\mathbf{C}(\Theta)+\sigma_\varepsilon^2 I\right)^{-1} \frac{\partial\left(\mathbf{C}(\Theta)+\sigma_\varepsilon^2 I\right)}{\partial \theta_i}\left(\mathbf{C}(\Theta)+\sigma_\varepsilon^2 I\right)^{-1} y.\end{aligned} \tag{2.27}$$

Minimisation of the negative loglikelihood can be done using any standard optimisation technique such as gradient descent. We will not focus on such methods here, but instead refer the interested reader to the optimisation chapter in [47] as an introduction, or to [58] for a more detailed exposition.

2.2. The Regularisation Approach in the Context of Inverse Problems

Having presented the Bayesian framework to multivariate function approximation we now turn to the classical approach. In fact, inference of a smooth function can also be explored in the context of inverse problems. As references for the material covered in this section refer to [14], [29], [24].

2.2.1. Tikhonov Regularisation

Let H_1, H_2 denote separable Hilbert spaces with respective norms $\|\cdot\|_{H_1}, \|\cdot\|_{H_2}$ and let $\mathcal{A} : H_1 \to H_2$ be a linear continuous operator. Consider the operator equation

$$\mathcal{A}f = g, \tag{2.28}$$

where $f \in H_1$ and $g \in H_2$. The *direct* problem consists of finding the output g for given input f, whereas the *inverse* problem is to find the input f that solves (2.28) for given g. The latter is called *well-posed* in the sense of Hadamard, if the solution f exists, is unique and is stable under slight corruption of g, meaning that f_1, f_2 are close if g_1, g_2 are close.

In practice, one often does not know g exactly but one has to make do with noisy data g^δ. Then, it can happen that $g^\delta \notin \mathcal{R}(\mathcal{A})$, meaning the problem is *ill-posed* as it has no solution. In order to overcome this, one generalises the problem by minimising the defect

$$\|\mathcal{A}f - g\|_{H_2} \to \min. \tag{2.29}$$

When minimising in H_1, however, the solution hereof does not have to be unique. One therefore wants to restrict the minimisation to a subset $V \subset H_1$ in order to enforce uniqueness. A possible definition of this set V can be made using an operator $\Omega : H_1 \to H_2$ describing desired features and demanding that the solution f fulfills this additional knowledge,

$$V := \{f \in H_1 : \Omega(f) = 0\}. \tag{2.30}$$

Here, we assume that the regularisation operator Ω is strictly convex on the null space of $\mathcal{A}$, $\mathcal{K}(\mathcal{A})$, and non-negative on H_1. Furthermore, we require it to be lower semi-continuous. Reformulating the restricted minimisation problem using Lagrangian multipliers leads to

$$\min_{f \in H_1} J_\lambda(f) = \min_{f \in H_1} \|\mathcal{A}f - g\|_{H_2}^2 + \lambda\Omega(f), \tag{2.31}$$

where $\lambda > 0$ is called the *regularisation strength*.

In order to derive a solution to the regularised minimisation problem (2.31) we specialise to

$$\Omega(f) := \|\mathcal{B}f\|_{H_3}^2, \tag{2.32}$$

with a linear operator $\mathcal{B} : H_1 \to H_3$ mapping onto a Hilbert space H_3 whose domain $\mathcal{D}(\mathcal{B})$ is dense in H_1. The operator $(\mathcal{B}^\star\mathcal{B}) : H_1 \to H_1$ is assumed to be strictly monotone on $\mathcal{K}(\mathcal{A})$, i.e. there exists $\beta > 0$ such that

$$\|\mathcal{B}f\|_{H_3} \geqslant \beta\|f\|_{H_1} \text{ for } f \in \mathcal{K}(\mathcal{A}). \tag{2.33}$$

This means that $\mathcal{B}$ can actually be set to zero on $\mathcal{K}(\mathcal{A})^\perp$, as $\mathcal{A}$ is non-negative there. Under these conditions the regularised minimisation functional,

$$J_\lambda(f) = \|\mathcal{A}f - g\|_{H_2}^2 + \lambda\|\mathcal{B}f\|_{H_3}^2, \tag{2.34}$$

is called *Tikhonov* functional and is minimised by the solution f_λ of the corresponding regularised normal equation,

$$(\mathcal{A}^\star\mathcal{A} + \lambda\mathcal{B}^\star\mathcal{B})f_\lambda = \mathcal{A}^\star g. \tag{2.35}$$

To see this, consider first the operator defined by the left hand side of the normal equation (2.35).

Lemma 1.
The operator $\mathcal{C} : H_1 \to H_1$ defined as $\mathcal{C} = \mathcal{A}^\star\mathcal{A} + \lambda\mathcal{B}^\star\mathcal{B}$ is positive definite, self-adjoint and injective.

Proof. First of all, consider the case $0 \neq f \in \mathcal{K}(\mathcal{A})$. Then it holds that

$$\langle \mathcal{C}f, f\rangle_{H_1} = \lambda\|\mathcal{B}f\|_{H_3}^2 \geqslant \lambda\beta^2\|f\|_{H_1}^2 > 0. \tag{2.36}$$

For $0 \neq f \in \mathcal{K}(\mathcal{A})^\perp$ we have with the above assumption concerning $\mathcal{B}$

$$\langle \mathcal{C}f, f\rangle_{H_1} = \|\mathcal{A}f - g\|_{H_2}^2 > 0. \tag{2.37}$$

Combining 2.36 and 2.37, we obtain that $\mathcal{C}$ is positive definite and thus injective. Secondly, for $f, \bar{f} \in H_1$ we have

$$\langle \mathcal{C}f, \bar{f} \rangle_{H_1} = \langle \mathcal{A}f, \mathcal{A}\bar{f} \rangle_{H_2} + \lambda \langle \mathcal{B}f, \mathcal{B}\bar{f} \rangle_{H_3} = \langle f, \mathcal{C}\bar{f} \rangle_{H_1}, \tag{2.38}$$

thus $\mathcal{C}$ is a self-adjoint operator. □

With these properties, one can now formally prove the above statement.

Theorem 1.
Let $\mathcal{A} : H_1 \to H_2$ be a linear continuous operator between Hilbert spaces, and let $\lambda > 0$. Then J_λ has a unique minimum. It is given by the solution $f_\lambda := \mathcal{C}^{-1}\mathcal{A}^\star g$ of the regularised normal equation (2.35).

Proof. Let $(f_n) \subset H_1$ be a minimizing sequence, i. e. we have

$$J_\lambda(f_n) \to \inf_{f \in H_1} J_\lambda(f) =: J. \tag{2.39}$$

Then

$$\begin{aligned} J_\lambda(f_n) + J_\lambda(f_m) &= 2J_\lambda(\frac{1}{2}(f_n + f_m)) \\ &\quad + \frac{1}{2}\|\mathcal{A}(f_n - f_m)\|^2_{H_2} + \frac{\lambda}{2}\|\mathcal{B}(f_n - f_m)\|^2_{H_3} \\ &\geqslant 2J + \frac{\lambda}{2}\beta^2 \|f_n - f_m\|^2_{H_1}, \end{aligned} \tag{2.40}$$

where the left hand side converges to $2J$. Thus (f_n) is a Cauchy sequence in a Hilbert space and hence it converges to $f_\lambda \in H_1$. Since J_λ is lower semi-continuous, it follows that

$$J = \liminf_{n\to\infty} J_\lambda(f_n) \geqslant J_\lambda(\liminf_{n\to\infty} f_n) = J_\lambda(f_\lambda). \tag{2.41}$$

This proves existence of the solution f_λ. Uniqueness of the solution follows from injectivity of $\mathcal{C}$.

To see that $\mathcal{A}^\star g$ lies in the range of the operator $\mathcal{C}$ and that hence $\mathcal{C}f_\lambda = \mathcal{A}^\star g$ can be uniquely solved for f_λ, one checks that (2.35) is actually the Euler-Lagrange equation of the Tikhonov functional (2.34).

Finally, in order to show that the solution f_λ minimises J_λ one calculates for $f \in H_1$ using Lemma 1 that

$$\begin{aligned} J_\lambda(f) &= \langle \mathcal{C}f, f \rangle_{H_1} - 2\langle Af, g \rangle_{H_2} + \|g\|^2_{H_2} \\ &= \langle \mathcal{C}f, f \rangle_{H_1} - 2\langle \mathcal{C}f_\lambda, f \rangle_{H_2} + \|g\|^2_{H_2} \\ &= \langle \mathcal{C}(f - f_\lambda), f - f_\lambda \rangle_{H_1} + \langle \mathcal{C}f_\lambda, f_\lambda \rangle_{H_2} + \|g\|^2_{H_2} \\ &\geqslant \lambda\beta^2 \|f - f_\lambda\|^2_{H_1} + \|g\|^2_{H_2} - \langle \mathcal{A}f_\lambda, g \rangle_{H_2}. \end{aligned} \tag{2.42}$$

□

2.2.2. Application to Function Inference

The goal now is to apply the theory of Tikhonov Regularisation to the task of infering a smooth function $f : \mathbb{R}^d \to \mathbb{R}$ from given noisy data $\mathcal{D} := (X, \mathbf{y}) = \{\mathbf{x}_n, y_n\}_{n=1,\dots,N}$, related via

$$y_n = f(\mathbf{x}_n) + \varepsilon, \tag{2.43}$$

where ε describes the noise. To do so consider the projection operator

$$\mathcal{P} : H_1 \to \mathbb{R}^N,\ \mathcal{P}f = \{f(\mathbf{x}_i)\}_{i=1}^N. \tag{2.44}$$

Then $\mathcal{P}$ is a linear and compact operator and H_1 is chosen as a suitable separable Hilbert space such that $\mathcal{P}$ is also continuous. Setting $\mathcal{A} = \mathcal{P}$ in the definition (2.34), the Tikhonov functional reads

$$J_\lambda(f) = \|\mathcal{P}f - y\|^2_{\mathbb{R}^N} + \lambda\|\mathcal{B}f\|^2_{H_3}. \tag{2.45}$$

A common choice for the inner product on $\mathbb{R}^N$ is the standard Euclidean inner product. Then one obtains the empirical quadratic error as the data dependent term, i.e.

$$J_\lambda(f) = \sum_{i=n}^{N}(f(\mathbf{x}_n) - y_n)^2 + \lambda\|\mathcal{B}f\|^2_{H_3}. \tag{2.46}$$

Application of Theorem 1 gives existence and uniqueness of the solution $f_\lambda = (\mathcal{P}^\star\mathcal{P} + \lambda\mathcal{B}^\star\mathcal{B})^{-1}\mathcal{P}^\star g$.

We now aim for a finite-dimensional representation of f_λ by taking advantage of the isometric isomorphism between H_1 and ℓ^2. To this end, denote by $\{\varphi(\mathbf{x})\}_{j=1}^\infty$ an orthonormal basis of H_1 to be chosen later and by $\gamma = \{\gamma_j\}_{j=1}^\infty$ the coefficients of the series expansion of the solution

$$f_\lambda(\mathbf{x}) = \sum_{j=1}^{\infty}\gamma_j\varphi_j(\mathbf{x}). \tag{2.47}$$

Then the operators $\mathcal{P}^\star\mathcal{P}$ and $\mathcal{B}^\star\mathcal{B}$ transform as follows

$$\begin{aligned}
\langle\mathcal{P}^\star\mathcal{P}f, f\rangle_{H_1} &= \sum_{k=1}^{\infty}\sum_{j=1}^{\infty}\gamma_j\langle\mathcal{P}\varphi_j, \mathcal{P}\varphi_k\rangle_2\gamma_k \\
&= \sum_{k=1}^{\infty}\sum_{j=1}^{\infty}\gamma_j\sum_{n=1}^{N}\varphi_j(\mathbf{x}_n)\varphi_k(\mathbf{x}_n)\gamma_k \\
&= \langle P^TP\gamma, \gamma\rangle_{l^2}, \\
\langle\mathcal{B}^\star\mathcal{B}f, f\rangle_{H_1} &= \sum_{k=1}^{\infty}\sum_{j=1}^{\infty}\gamma_j\langle\mathcal{B}^\star\mathcal{B}\varphi_j, \varphi_k\rangle\gamma_k \\
&= \langle B\gamma, \gamma\rangle_{l^2},
\end{aligned} \tag{2.48}$$

with infinite-dimensional matrices P and B where

$$P_{nj} = \varphi_j(\mathbf{x}_n) \text{ and } B_{ij} = \langle \mathcal{B}\varphi_i, \mathcal{B}\varphi_j \rangle_{H_1} \text{ for } n = 1, 2, ..., N,\ i, j = 1, 2, ... \quad (2.49)$$

Hence the coefficients γ solve the linear system

$$(P^T P + \lambda B)\gamma = P^T y. \quad (2.50)$$

While this formulation is already simpler, it is still infinite-dimensional. By choosing the orthonormal basis in a suitable way, however, it is possible to discard the infinite series expansion in favor of a finite one. W.l.o.g. assume that B is positive definite. This can always be achieved by defining

$$B = \tilde{\mathcal{B}}^T \tilde{\mathcal{B}}, \text{ with } \tilde{\mathcal{B}} = \sqrt{\mathcal{B}^T \mathcal{B}}. \quad (2.51)$$

Then we can define $\{\varphi(x)\}_{j=1}^{\infty}$ as eigenvectors of B - corresponding to the eigenvalues $\{\mu\}_{j=1}^{\infty}$ - as they form an orthonormal basis of H_1. Consequently the k-th row of (2.50) reads as

$$\sum_{i=1}^{\infty} \sum_{n=1}^{N} \gamma_i \varphi_k(\mathbf{x}_n)\varphi_i(\mathbf{x}_n) + \lambda \mu_k \gamma_k = \sum_{n=1}^{N} \varphi_k(\mathbf{x}_n) y_n. \quad (2.52)$$

Combining the infinite sum to $f(\mathbf{x}_n)$, one obtains for the coefficients

$$\gamma_k = \frac{1}{\lambda \mu_k} (\sum_{n=1}^{N} \varphi_k(\mathbf{x}_n)(y_n - f(\mathbf{x}_n))), \quad (2.53)$$

which can be plugged into the series expansion of f to give

$$f(\mathbf{x}) = \sum_{n=1}^{N} (\sum_{k=1}^{\infty} \frac{1}{\mu_k} \varphi_k(\mathbf{x})\varphi_k(\mathbf{x}_n)) \frac{1}{\lambda}(y_n - f(\mathbf{x}_n)) = \sum_{n=1}^{N} \alpha_n \mathcal{k}(\mathbf{x}, \mathbf{x}_n), \quad (2.54)$$

with

$$\mathcal{k}(x, y) = \sum_{k=1}^{\infty} \frac{1}{\mu_k} \varphi_k(x)\varphi_k(y) \text{ and } \alpha_n = \frac{1}{\lambda}(y_n - f(\mathbf{x}_n)). \quad (2.55)$$

Hence f can be represented as a finite sum of data-dependent functions $\mathcal{k}(\cdot, \mathbf{x}_n)$ with coefficients α solving the finite-dimensional linear system

$$(\mathcal{K} + \lambda I)\alpha = \mathbf{y}, \quad (2.56)$$

where $\mathcal{K} = \{\mathcal{k}(\mathbf{x}_i, \mathbf{x}_j)\}_{i,j=1}^{N}$ is a symmetric and positive semidefinite matrix. This can be seen by calculating

$$\begin{aligned} \sum_{i,j=1}^{N} c_i \mathcal{K}_{ij} c_j &= \sum_{i,j=1}^{N} c_i \sum_{k=1}^{\infty} \frac{1}{\mu_k} \psi_k(x_i)\psi_k(x_j) c_j \\ &= \sum_{k=1}^{\infty} \frac{1}{\mu_k} \langle c, \psi_k(X) \rangle_{\mathbb{R}^N}^2 \geqslant 0. \end{aligned} \quad (2.57)$$

Thus the function $\mathcal{k}$ is in fact a so-called kernel, cf. Definition 2.

The Feature Space

One can now take a closer look at the space

$$\mathcal{H} = \overline{\text{span}\{\sum_{i=1}^{m} \alpha_i k(\cdot, \mathbf{x}_i') | m \in \mathbb{N}, \alpha_i \in \mathbb{R}, \mathbf{x}_1', ..., \mathbf{x}_m' \in \mathbb{R}^d\}}$$

in which the solution f is found. In fact, we will observe that this is exactly the feature space associated with the kernel k as noted in section 2.1.1.

Defining a dot product on $\mathcal{H}$ as

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{i=1}^{m} \sum_{j=1}^{m'} \alpha_i \beta_j k(\mathbf{x}_i, \mathbf{x}_j') \tag{2.58}$$

for $f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, x_i)$ and $g(\cdot) = \sum_{j=1}^{m'} \beta_j k(\cdot, x_j')$, $\mathcal{H}$ becomes a Hilbert space with k acting as the representer of evaluation, since

$$\langle k(\cdot, \mathbf{x}'), f \rangle = \sum_{i=1}^{m} \alpha_i k(x', x_i) = f(x'). \tag{2.59}$$

In particular, pointwise evaluations are bounded, as

$$|f(\mathbf{x})|^2 = |\langle k(\cdot, \mathbf{x}), f \rangle|^2 \leqslant k(\mathbf{x}, \mathbf{x}) \|f\|. \tag{2.60}$$

This motivates the following definition.

Definition 3.
A reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ *is defined as a Hilbert space of real functions* $f : \mathcal{X} \to \mathbb{R}$ *such that all evaluating functionals* $f \mapsto f(x)$, $x \in \mathcal{X}$, *are continuous.*

It was shown above that one can construct a RKHS from a given positive definite function k. On the other hand, as indicated by the name, the existence of a reproducing kernel for a given Hilbert space follows by the Riesz representation Theorem,

$$\forall x \in \mathcal{X} : \exists! k(\cdot, x) \text{ such that } f(x) = \langle f, k(\cdot, x) \rangle, \tag{2.61}$$

and the definition

$$k(x, x') = \langle k(\cdot, x), k(\cdot, x') \rangle_{\mathcal{H}}. \tag{2.62}$$

Thus for every regularisation operator Ω of the form $\Omega(f) = \|\mathcal{B}f\|^2$ there exists a RKHS with a kernel k such that the Tikhonov functional (2.46) can be equivalently written as

$$J_\lambda(f) = \sum_{i=n}^{N} (f(\mathbf{x}_n) - y_n)^2 + \lambda \|f\|_{\mathcal{H}}^2, \tag{2.63}$$

and the solution f can be represented as a finite linear combination of kernels centered on the data points. This can be stated in a more general context for arbitrary loss functions instead of the empirical quadratic error. The following theorem is taken from [47], see therein for the proof. As before, $\mathcal{H}$ denotes a RKHS with its reproducing kernel k.

Theorem 2 (Representer Theorem)**.**
Denote by $\Omega : [0, \infty) \to \mathbb{R}$ a strictly monotone increasing function, by $\mathcal{X}$ a set, and by $c : \mathcal{X} \times \mathbb{R}^2 \to [0, \infty)$ an arbitrary loss function, i. e., $c(x, y, y) = 0$ for all $x \in \mathcal{X}$ and $y \in \mathbb{R}$. Then each minimizer $f \in \mathcal{H}$ of the regularised risk

$$c((x_1, y_1, f(x_1)), ..., (x_m, y_m, f(x_m))) + \Omega(\|f\|_{\mathcal{H}}) \tag{2.64}$$

admits a representation of the form

$$f(x) = \sum_{i=1}^{m} \alpha_i \mathcal{k}(x_i, x). \tag{2.65}$$

This result allows for the treatment of many different kernel-based machine learning methods under one generalised framework. The respective methods differ in the choice of the specific loss function and regularisation operator. The standard support vector regression (c.f. [47]), for example, corresponds to the use of the so-called ε-insensitive loss function,

$$c(x, y, f(x)) = \max\{0, |y - f(x)| - \varepsilon\}, \tag{2.66}$$

and the regularisation operator $\Omega(||f||_{\mathcal{H}}) = \frac{1}{2}||f||^2_{\mathcal{H}}$.

In order to establish a rigorous connection between regularisation operator Ω and kernel k, the question remains whether for every RKHS there exists a corresponding regularisation operator of the form $||\mathcal{B} \cdot ||_{H_3}$. It is answered by the next theorem.

Theorem 3.
For every RKHS $\mathcal{H}$ with reproducing kernel $\mathcal{k}$ there exists a linear operator $\mathcal{B} : \mathcal{H} \to H_3$ such that for all $f \in \mathcal{H}$,

$$\langle \mathcal{B}\mathcal{k}(x, \cdot), \mathcal{B}f(\cdot) \rangle = f(x), \tag{2.67}$$

and in particular,

$$\langle \mathcal{B}\mathcal{k}(x, \cdot), \mathcal{B}\mathcal{k}(x', \cdot) \rangle = \mathcal{k}(x, x') \tag{2.68}$$

However, the correspondence between regularisation operator and kernel does not have to be unique. For example choosing $\mathcal{B}$ as the identity and H_3 as $\mathcal{H}$ always provides a valid regularisation operator, effectively proving the theorem. Typically, the difficulty lies in finding a regularisation operator corresponding to a specific Hilbert space H_3.

2.3. Relation between Regularisation and GP Regression

As we have seen in the sections before, the equations (2.15) and (2.56) stating the solutions to GP regression and Tikhonov regularisation are identical. In this section, the connection between the two approaches is illustrated. For further reading refer to [42]. We note that, in the Machine Learning community, the approach of deriving Equation (2.56) via Tikhonov regularisation is also called *kernel ridge regression (KRR)*.

Applying the exponential map to the negative regularised Tikhonov functional (2.46), one obtains

$$\exp\left(-\frac{1}{\lambda}J_\lambda(f)\right) = \exp\left(-\frac{1}{\lambda}\sum_{n=1}^{N}(f(x_n)-y_n)^2\right)\exp\left(-\|\mathcal{B}f\|^2\right). \tag{2.69}$$

Here the first term on the right hand side is proportional to the likelihood of the GP regression

$$p(y|f) = \mathcal{N}(f, \sigma_\varepsilon^2 I) \tag{2.70}$$

if one identifies λ with the variance σ_ε^2.

The second term corresponds to the prior of the GP regression (2.8). To see this, we use the expansion of the latent training values in the kernel functions,

$$f(x_i) = \sum_{j=1}^{N}\alpha_j \mathcal{k}(x_j, x_i), \tag{2.71}$$

to calculate

$$\begin{aligned} p(f|X) \sim \exp\left(-f(X)^T C^{-1} f(X)\right) &= \exp\left(-\alpha^T C\alpha\right) \\ &= \exp\left(-\|f\|_{\mathcal{H}}^2\right) = \exp\left(-\|\mathcal{B}f\|^2\right). \end{aligned} \tag{2.72}$$

Here, we have used that $C_{ij} := \mathcal{k}(x_i, x_j)$ and we have chosen $\mathcal{B}$ as the regularisation operator corresponding to the kernel $\mathcal{k}$. This means that by choosing a regularisation operator $\mathcal{B}$ one selects a particular prior, even when not thinking in the bayesian framework.

We can now conclude that since $f^\star$ minimises $J_\lambda(f)$, it maximises (2.69) and hence it is the *maximum a posteriori* (MAP) solution. For a Gaussian distribution the mode and the mean coincide. Therefore, the solution of the regularised Tikhonov functional and the prediction of the Gaussian process regression are equal. Note however, that for this equivalence to hold the assumption it is crucial to assume that both data and noise are Gaussian distributed.

3. Application to the Potential Energy Hypersurface

In this chapter we apply the mathematical theory of Gaussian process regression to the approximation of the high-dimensional Born-Oppenheimer potential energy surface. We start by describing the concept of this surface as well as the fundamental energy decomposition the framework is based on. We continue by deriving an explicit functional representation for the atomic contributions to the total energy and extend the prediction to gradient values. These are needed in molecular dynamics simulations as they correspond to the negative forces acting on the particles. We also describe how to incorporate gradient information into the training data in order to enhance function value prediction.

We conclude this chapter by presenting an extension of the framework to the learning of energy correction terms.

3.1. The Born-Oppenheimer Potential Energy Surface

As a reference to the subject from the point of view of computational chemistry we note the book by Errol G. Lewars, [27], Chapter 2.

The potential energy surface is a hypothetical concept relating the geometric structure of a (finite) chemical compound to its energy. It usually neglects that the atoms composing the molecule are not stationary but can occupy different vibrational energy levels depending on the ambient temperature.

It is motivated by the Born-Oppenheimer approximation of quantum mechanics. This approximation postulates that due to their huge difference in mass the movements of the nucleii and the electrons occur on different time scales. This means that the electrons effectively see the nuclei as stationary. Mathematically, this assumption allows the Schroedinger equation to become decoupled into an electronic and a nuclear equation, where the electronic one depends no longer on time but only on the nuclear coordinates. This electronic potential defines the potential energy surface on which the nuclei move according to the laws of classical mechanics.

Intuitively, this allows us to think of the potential energy surface as the graph of the function that maps nuclear coordinates to energy,

$$E = E(\mathbf{x}_1, ..., \mathbf{x}_P). \tag{3.1}$$

The resulting high-dimensional hypersurface can be used for a descriptive interpretation of chemical reactions. Minima of the surface correspond to chemical configurations having a lower energy than any surrounding configuration that could be reached by slightly perturbing the geometric coordinates. Chemically speaking, this is the definition of a meta-stable state. First-order saddle points on the other hand are configurations for which the energy is minimal under perturbation in all directions but one. Changing the coordinates in this direction leads to configurations with a lower energy, meaning one has identified a transition state. This way the PES can be searched for meta-stable configurations and the reaction paths between them.

For their ability to introduce mathematical techniques to the analysis of chemical problems, PES are an important construct in computational chemistry. Their accurate calculation within the limits of the Born-Oppenheimer approximation requires the solving of the electronic Schroedinger equation. While already simplified, the numerical calculation of the potential is still computationally demanding. Hence, one aims to restrict the evaluation of the approximated Schroedinger equation to a minimum. This justifies the large interest in interpolating the PES using techniques as presented in the preceding chapter.

We will now continue with the application of the Gaussian process regression to PES interpolation.

3.2. The Atomic Decomposition Ansatz

It is our aim to present a framework for interpolating potential energy hypersurfaces needed in the molecular dynamics simulations of general particle systems. They should be applicable to molecules and crystalline solids alike.

When considering the total energy of a P-particle system, we will assume it can be written as the sum of the P atomic contributions,

$$E_{\text{total}} = \sum_{p=1}^{P} E_{\text{atomic}}(q_p), \tag{3.2}$$

where q_p describes the representation of the neighbourhood of the p-th atom using some generalised characteristics. For molecular dynamics applications, it is these atomic contributions one needs to evaluate. Quantum mechanical calculations, however, are only able to provide the total energy E_{total}. Hence, it is our goal to use Gaussian process regression based on the observed total energy values $\mathbf{y}$ in order to derive an expression for the atomic energy contributions $E_{\text{atomic}}(q_p)$. In this chapter the target function to be infered is the atomic energy, which we will denote by $f(q) = E_{\text{atomic}}(q)$.

In order to be able to cope with very large particle systems or even infinite periodic structures like crystals, we have to limit the atomic neighbourhoods to a finite region by introducing a cut-off radius, effectively breaking the system

into smaller subsystems. Strictly speaking this only makes sense if the following localisation assumption holds:

Assumption 1 (Localisation)**.**
The atomic contribution depends only on the environment q within a suitable cut-off radius of the atom.

If the assumption is violated for the total energy, it is always possible to learn only short range interactions via regression, which are localised by definition, and let the long-term contributions be modeled by an explicit potential. We will address this issue in Section 3.4.

3.2.1. Deriving an Expression for Atomic Energy Contributions

We now apply the atomic decomposition ansatz to the Gaussian process regression described in Section 2.1.1 in order to derive an expression for the inference of the atomic energy function, $f(q) = E_{\text{atomic}}(q)$. The concept of localised Gaussian process regression for the interpolation of the PES was introduced by Bartók *et al.* under the name of Gaussian Approximation Potentials (GAP) [4, 1]. However, they do not describe the aspect of the atomic contributions in much detail, which is why we will recapitulate the derivation.

Let N denote the number of training examples, i.e. we have N particle systems with N corresponding total energy values $\mathbf{y}$ where system n consists of P_n particles. As usual, we assume that the observed energy values differ from the underlying energy function E_{total} by additive, i.i.d. Gaussian noise ε, i. e.,

$$\mathbf{y} = \mathbf{E}_{\text{total}} + \varepsilon, \text{ with } \varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2). \tag{3.3}$$

Denote by $\mathbf{f} = \{E_{\text{atomic}}(q_k)\}_{k=1,\dots,K}$ the $K := \sum_{n=1}^{N} P_n$ local and unobservable atomic contributions, where q_k describes the local environment of the k-th atom. Let L be the $K \times N$ matrix relating the total to the atomic energies, i. e.,

$$\mathbf{E}_{\text{total}} = L^T \mathbf{f}. \tag{3.4}$$

Then we can relate the covariance matrix $\mathbf{C}_N$ of the total energy values to the covariance matrix $\mathbf{C}_K$ of the atomic contributions as

$$\begin{aligned}(\mathbf{C}_N)_{ij} := \text{Cov}(E_{\text{total}}(\mathbf{x}_i), E_{\text{total}}(\mathbf{x}_j)) &= \text{Cov}(\sum_{l=1}^{K} L_{li} f_l, \sum_{k=1}^{K} L_{kj} f_k) \\ &= \sum_{k,l=1}^{K} L_{li} (\mathbf{C}_K)_{l,k} L_{kj} \\ &= (\mathbf{L}^T \mathbf{C}_K \mathbf{L})_{ij}.\end{aligned} \tag{3.5}$$

Analogously, the covariance between the training data and a new test input $\mathbf{x}^\star$

with $P^\star$ particles can be written as

$$\begin{aligned}(\mathbf{k}_\star)_n := \mathrm{Cov}(E_{\mathrm{total}}(\mathbf{x}_n), E^\star_{\mathrm{total}}) &= \mathrm{Cov}(\sum_{l=1}^{K} L_{ln} f_l, \sum_{p=1}^{P^\star} f^\star_p) \\ &= \sum_{l=1}^{K} L_{ln} \sum_{p=1}^{P^\star} \mathrm{Cov}(f_l, f^\star_p) \\ &= \sum_{p=1}^{P^\star} (L^T \mathbf{c}^\star_{(p)})_n.\end{aligned} \tag{3.6}$$

As an expression for the variance of the test input we obtain

$$\mathbf{k}_{\star\star} := \mathrm{Cov}(E^\star_{\mathrm{total}}, E^\star_{\mathrm{total}}) = \sum_{p,p'=1}^{P^\star} \mathrm{Cov}(f^\star_p, f^\star_{p'}) = \sum_{p,p'=1}^{P^\star} (c^{\star\star})_{pp'}. \tag{3.7}$$

Thus the covariance between total energy values is decomposed, allowing for the definition of a kernel function based only on the representation of the local environment of a particle,

$$\begin{aligned}(\mathbf{C}_K)_{ij} &:= \mathcal{k}(q_i, q_j), \;\; i,j = 1,...,K, \\ (\mathbf{c}^\star_{(p)})_k &:= \mathcal{k}(q_k, q^\star_p), \;\; k = 1,...,K, \;\; p = 1,...,P^\star \\ (c^{\star\star})_{pp'} &:= \mathcal{k}(q^\star_p, q^\star_p), \;\; p,p' = 1,...,P^\star.\end{aligned} \tag{3.8}$$

As noted before, this is important in order to be able to cope with very large particle systems. In practice, we will use the Gaussian kernel as defined in Equation (2.17) as the standard choice.

Plugging the expressions into regular GP regression as described in Section 2.1.1 gives

$$\begin{aligned}\mathbb{E}(E^\star_{\mathrm{total}}) &= \sum_{p=1}^{P^\star} (\mathbf{c}^\star_{(p)})^T L (L^T \mathbf{C}_K L + \sigma_\varepsilon^2 I)^{-1} \mathbf{y}, \\ \mathbb{V}\mathrm{ar}(E^\star_{\mathrm{total}}) &= \sum_{p,p'=1}^{P^\star} \Big[(c^{\star\star})_{pp'} - (\mathbf{c}^\star_{(p)})^T L (L^T \mathbf{C}_K L + \sigma_\varepsilon^2 I)^{-1} L^T \mathbf{c}^\star_{(p')} \Big],\end{aligned} \tag{3.9}$$

as an estimate for the total energy of the input system $\mathbf{x}^\star$ and its variance based only on local atomic environments. The individual atomic contribution of the p-th atom of $\mathbf{x}^\star$ is

$$f(q^\star_p) = E^\star_{\mathrm{atomic}}(q^\star_p) = \sum_{k=1}^{K} \mathcal{k}(q_k, q^\star_p) \alpha_k, \tag{3.10}$$

with the coefficients $\alpha = L(L^T \mathbf{C}_K L + \sigma_\varepsilon^2 I)^{-1} \mathbf{y}$. As with regular Gaussian process regression, the interpolated function is reconstructed as a sum of weighted kernel functions centered at the training data, which now corresponds to the atomic environments instead of the complete particle system.

3.2.2. Relating Global to Local Indexing

In order to make the relation between global and local indexing clearer, we explicitly execute the localised Gaussian process regression for a small (theoretical) example.

Assume we have 2 training systems, the first with 3 particles and the other with 2. Hence, we need in total 3+2 = 5 local atomic environment representations q_k, $k = 1, ..., 5$. If we enumerate the local particles sequentially, the representations q_1 to q_3 correspond to the first particle system and q_4 and q_5 to the second. The matrix L relating total energy values to their local energy contributions therefore takes the form

$$L^T = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}. \tag{3.11}$$

By defining $k_{ij} := \mathcal{k}(q_i, q_j) = (\mathbf{C}_K)_{ij}$ we obtain

$$L^T \mathbf{C}_K L = \begin{pmatrix} \sum_{i,j=1}^{3} k_{ij} & \sum_{i=1}^{3}\sum_{j=4}^{5} k_{ij} \\ \sum_{i=1}^{3}\sum_{j=4}^{5} k_{ij} & \sum_{i,j=4}^{5} k_{ij} \end{pmatrix}, \tag{3.12}$$

and consequently,

$$(L^T \mathbf{C}_K L + \sigma_\varepsilon^2 I)^{-1} = \frac{1}{\Delta} \begin{pmatrix} \sum_{i,j=4}^{5} k_{ij} + \sigma_\varepsilon^2 & -\sum_{i=1}^{3}\sum_{j=4}^{5} k_{ij} \\ -\sum_{i=1}^{3}\sum_{j=4}^{5} k_{ij} & \sum_{i,j=1}^{3} k_{ij} + \sigma_\varepsilon^2 \end{pmatrix}, \tag{3.13}$$

where

$$\Delta := \det(L^T \mathbf{C}_K L + \sigma_\varepsilon^2 I) = (\sum_{i,j=1}^{3} k_{ij} + \sigma_\varepsilon^2)(\sum_{i,j=4}^{5} k_{ij} + \sigma_\varepsilon^2) - (\sum_{i=1}^{3}\sum_{j=4}^{5} k_{ij})^2.$$

Computing α as

$$\alpha_1 = \frac{1}{\Delta}(y_1(\sum_{i,j=4}^{5} k_{ij} + \sigma_\varepsilon^2) - y_2 \sum_{i=1}^{3}\sum_{j=4}^{5} k_{ij}) \tag{3.14}$$

$$\alpha_2 = \frac{1}{\Delta}(-y_1 \sum_{i=1}^{3}\sum_{j=4}^{5} k_{ij} + y_2(\sum_{i,j=1}^{3} k_{ij} + \sigma_\varepsilon^2)), \tag{3.15}$$

the final expression for the p-th local energy of an arbitrary system is given by

$$\begin{aligned} f(q_p^\star) = \alpha_1 k(q_1, q_p^\star) + \alpha_1 k(q_2, q_p^\star) + \alpha_1 k(q_3, q_p^\star) \\ + \alpha_2 k(q_4, q_p^\star) + \alpha_2 k(q_5, q_p^\star). \end{aligned} \tag{3.16}$$

3.3. Incorporating Derivatives

Aside from the local energy contributions, the molecular dynamics applications also need to compute the forces acting on each particle, i.e. the negative gradient of the potential with respect to the coordinates of the particles. Therefore, one wants to extend the Gaussian process regression to include predictions for the gradient. Furthermore, one can use information on the gradient to enhance the prediction of function values.

As before, let $\mathcal{D} = (X, \mathbf{y}) = \{\mathbf{x}_n, y_n\}_{n=1,...,N}$ denote the training data, with $\mathbf{x}_n$ representing the cartesian coordinates of the n-th particle system and y_n the corresponding observed energy value. The target energy values differ from the latent function values by i.i.d. Gaussian noise ε,

$$y_n = E_{\text{total}}(\mathbf{x}_n) + \varepsilon, \;\; n = 1, ..., N, \;\text{ with } \varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2). \tag{3.17}$$

In order to facilitate the notation, we will now assume w.l.o.g. that each of the training particle systems has exactly P atoms. This can always be achieved by introducing dummy atoms, whose influence is eliminated by the local environment descriptors. For the use of the local Coulomb matrix this is the case when assigning them a zero nuclear charge along with arbitrary cartesian coordinates.

Additionally, we assume that for each training system $\mathbf{x}_n \in \mathbb{R}^{d*P}$ the gradient is given by

$$g_n^k = \frac{\partial E_{\text{total}}(\mathbf{x}_n)}{\partial x_n^k}, \;\; n = 1, ..., N, \;\; k = 1, ..., d * P. \tag{3.18}$$

The gradient is afflicted with i.i.d. Gaussian noise κ,

$$\mathbf{z}_n = \mathbf{g}_n + \kappa, \;\; n = 1, ..., N, \;\text{ with } \kappa \sim \mathcal{N}(0, \sigma_\kappa^2). \tag{3.19}$$

Note that $\kappa \neq \varepsilon$, since in general the error of the gradient is one order of magnitude worse than the error of the function values.

3.3.1. Prediction of Gradient Values

Assume we have a test system $\mathbf{x}_\star$ with $P^\star$ particles and atomic environments $q_p^\star$, $p = 1, ..., P^\star$. In order to infere gradient values from training data including only function values, i.e. the energy observations but no forces, it suffices to differentiate the function reconstructed by the localised GP regression in Equation (3.10). It holds

$$g_\star^i = \frac{\partial E_{\text{total}}^\star}{\partial x_\star^i} = \sum_{p=1}^{P^\star} \sum_{k=1}^{K} \frac{\partial}{\partial x_\star^i} k(q_k, q_p^\star)\alpha_k, \;\text{ with } \alpha = \underbrace{L(L^T \mathbf{C}_K L + \sigma_\varepsilon^2 I)^{-1}\mathbf{y}}_{\text{independent of test system}}. \tag{3.20}$$

Analogously to the reconstructed energy function, we can write the gradient as a sum of weighted basis functions where the coefficients are the same as learned

by the regression for predicting the energy values and hence do not have to be calculated again. The basis functions are now derivatives of the local kernels with respect to the cartesian coordinates. Application of the chain rule (cf. Appendix C.1) leads to

$$\frac{\partial}{\partial x_\star^i} k(q_k, q_j^\star) = \frac{\partial k(q_k, q_j^\star)}{\partial q_j^\star} \cdot \frac{\partial q_j^\star}{\partial x_\star^i} = \frac{\partial k(q_k, q_j^\star)}{\partial \mathbf{q}^\star} \cdot \frac{\partial \mathbf{q}^\star}{\partial x_\star^i}, \tag{3.21}$$

Aside from the definition of the local environment representations $\mathbf{q}$ and $\mathbf{q}^\star$, one consequently also needs to specify the derivative of the representation with respect to the atomic coordinates, $\frac{\partial \mathbf{q}^\star}{\partial x_i^\star}$.

3.3.2. Inclusion of Gradients to Enhance Function Value Prediction

One can make use of available gradient data to enhance the prediction of function and gradient values for a test system. To this end, it suffices to apply Gaussian process regression to an extended target vector including the gradients,

$$\mathbf{y}_{\text{ext}} = (y_1, ..., y_N, \mathbf{z}_1, ..., \mathbf{z}_N)^T, \tag{3.22}$$

leading to the equation

$$\mathbb{E}(f^\star, \mathbf{g}^\star) = (\mathbf{c}_{\text{ext}}^\star)^T (\mathbf{C}_{\text{ext}} + I_{\text{ext}})^{-1} \mathbf{y}_{\text{ext}}. \tag{3.23}$$

Here, $\mathbf{C}_{\text{ext}}$ denotes an extended covariance matrix, comprising not only the covariance between different function values, but also between function and gradient values, as well as between different gradient values. It can be decomposed into

$$\mathbf{C}_{\text{ext}} = \begin{pmatrix} L^T \mathbf{C}_K L & \mathbf{C}_{f,g_1} & \dots & \mathbf{C}_{f,g_N} \\ \mathbf{C}_{g_1,f} & \mathbf{C}_{g_1,g_1} & \dots & \mathbf{C}_{g_1,g_N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{g_N,f} & \mathbf{C}_{g_N,g_1} & \dots & \mathbf{C}_{g_N,g_N} \end{pmatrix}, \tag{3.24}$$

with

$$\begin{aligned} \mathbf{C}_{f,g_n} &= \{\text{Cov}(E_{\text{total}}(\mathbf{x}_i), \frac{\partial E_{\text{total}}(\mathbf{x}_n)}{\partial x_n^j})\}_{i=1,...,N; j=1,...,d*P_n}, \\ \mathbf{C}_{g_n,g_m} &= \{\text{Cov}(\frac{\partial E_{\text{total}}(\mathbf{x}_n)}{\partial x_n^i}, \frac{\partial E_{\text{total}}(\mathbf{x}_m)}{\partial x_m^j})\}_{i=1,...,d*P_n; j=1,...,d*P_m}. \end{aligned} \tag{3.25}$$

Analogously, the $\mathbf{c}_{\text{ext}}^\star$ is comprised of the covariances between the extended data and the test system,

$$\mathbf{c}_{\text{ext}}^\star = \begin{pmatrix} \mathbf{C}_{f^\star,f}^\star & \mathbf{C}_{f^\star,g_1}^\star & \dots & \mathbf{C}_{f^\star,g_N}^\star \\ \mathbf{C}_{g^\star,f}^\star & \mathbf{C}_{g^\star,g_1}^\star & \dots & \mathbf{C}_{g^\star,g_N}^\star \end{pmatrix}, \tag{3.26}$$

where

$$\begin{aligned}
\mathbf{C}^{\star}_{f^{\star},f} &= \{\mathrm{Cov}(E^{\star}_{\mathrm{total}}, E_{\mathrm{total}}(\mathbf{x}_n))\}_{n=1,\ldots,N},\\
\mathbf{C}^{\star}_{f^{\star},g_n} &= \{\mathrm{Cov}(E^{\star}_{\mathrm{total}}, \frac{\partial E_{\mathrm{total}}(\mathbf{x}_n)}{\partial x^k_n})\}_{k=1,\ldots,d*P},\\
\mathbf{C}^{\star}_{g^{\star},f} &= \{\mathrm{Cov}(\frac{\partial E^{\star}_{\mathrm{total}}}{\partial x^i_{\star}}, E_{\mathrm{total}}(\mathbf{x}_n))\}_{i=1,\ldots,d*P^{\star};n=1,\ldots,N},\\
\mathbf{C}^{\star}_{g^{\star},g_n} &= \{\mathrm{Cov}(\frac{\partial E^{\star}_{\mathrm{total}}}{\partial x^i_{\star}}, \frac{\partial E_{\mathrm{total}}(\mathbf{x}_n)}{\partial x^k_n})\}_{i=1,\ldots,d*P^{\star};k=1,\ldots,d*P},
\end{aligned} \tag{3.27}$$

and the matrix I_{ext} deals with the implications of noisy data,

$$I_{\mathrm{ext}} = \begin{pmatrix} \sigma^2_{\varepsilon} I_N & \mathbf{0} \\ \mathbf{0} & \sigma^2_{\kappa} I_{d*K} \end{pmatrix}. \tag{3.28}$$

The entries in the extended covariance matrices $\mathbf{C}_{\mathrm{ext}}$ and $\mathbf{c}^{\star}_{\mathrm{ext}}$ can be calculated rigorously (see Appendix C.2 for the details).

As before, both function value and gradients can be written as a weighted sum of basis functions centered on the data points. The weights are the extended coefficients learnt by the Gaussian process regression,

$$\alpha_{\mathrm{ext}} = (\mathbf{C}_{\mathrm{ext}} + I_{\mathrm{ext}})^{-1}\mathbf{y}_{\mathrm{ext}}, \tag{3.29}$$

which can be partitioned according to the structure of the extended covariance matrix $\mathbf{C}_{\mathrm{ext}}$ into

$$\alpha = (\beta_1, ..., \beta_N, \gamma_1, ..., \gamma_N)^T, \text{ with } \beta_n \in \mathbb{R} \text{ and } \gamma_n \in \mathbb{R}^{d*P}. \tag{3.30}$$

Then, identifying the mean value of the posterior predictive distribution with the prediction for the total energy of the test system as usual, we obtain

$$\begin{aligned}
E^{\star}_{\mathrm{total}} &= C^{\star}_{f^{\star},f}\beta + \sum_{n=1}^{N} \mathbf{C}^{\star}_{f^{\star},g_n}\gamma_n \\
&= \sum_{p=1}^{P^{\star}} (\mathbf{c}^{\star}_{(p)})^T \mathbf{L}\beta + \sum_{n=1}^{N} \mathbf{L}^T_{n,:}\mathbf{d}^n_{(p)}\gamma_n.
\end{aligned} \tag{3.31}$$

For the local energy of the p-th atom this gives

$$f(q^{\star}_p) = \sum_{k=1}^{K} \beta'_k \mathcal{k}(q_k, q^{\star}_p) + \sum_{k=1}^{K}\sum_{n=1}^{N}\sum_{i=1}^{d*P} (\gamma'_{ni})^{(k)} \frac{\partial \mathcal{k}(q_k, g^{\star}_p)}{\partial \mathbf{q}} \cdot \frac{\partial \mathbf{q}}{\partial x^i_n}, \tag{3.32}$$

where

$$\begin{aligned}
\beta' &= \mathbf{L}\beta, \\
(\gamma'_{nk})^{(i)} &= L_{in}\gamma^{(k)}_n.
\end{aligned} \tag{3.33}$$

The basis set in which the reconstructed function is expanded is now comprised of both the kernel functions centered at the training environments, $\mathcal{k}(q_i, q_p^\star)$, and the corresponding derivatives, $\frac{\partial \mathcal{k}(q_i, g_p^\star)}{\partial \mathbf{q}}$.

Analogously to the preceding subsection, one can now differentiate this expression with respect to the cartesian coordinates of the test system in order to obtain a prediction for the gradient. As before, the coefficients are transfered, since they are independent of the test system. The basis set transforms to the first and second order derivatives of the local kernel $\mathcal{k}$,

$$\begin{aligned} g_\star^j &= \sum_{p=1}^{P^\star} \frac{\partial}{\partial x_\star^j} f(q_p^\star) \\ &= \sum_{p=1}^{P^\star} \sum_{k=1}^{K} \beta_k' \frac{\partial \mathcal{k}(q_k, q_p^\star)}{\partial \mathbf{q}^\star} \cdot \frac{\partial \mathbf{q}^\star}{\partial x_\star^j} \\ &\quad + \sum_{p=1}^{P^\star} \sum_{k=1}^{K} \sum_{n=1}^{N} \sum_{i=1}^{d*P} (\gamma_{ni}')^{(k)} \Big(\frac{\partial \mathbf{q}^\star}{\partial x_\star^j} \Big)^T \cdot \frac{\partial^2 \mathcal{k}(q_k, g_p^\star)}{\partial \mathbf{q} \partial \mathbf{q}^\star} \cdot \frac{\partial \mathbf{q}}{\partial x_n^i}. \end{aligned} \tag{3.34}$$

3.4. Extension to Learning Energy Differences

The GAP can also be used to learn energy correction terms. Assume that we have two models for the total energy, a costly but accurate model A and a cheap but not so accurate model B. Then we use the GAP to calculate only the difference between the models,

$$E_{\mathrm{GAP}} = E_{\mathrm{Model\ A}} - E_{\mathrm{Model\ B}}. \tag{3.35}$$

A straigthforward application would be for example to choose model A as a DFT calculator and model B as an empirical potential whose accuracy we would like to improve. This can be done using any empirical potential. To obtain a prediction value for the total energy, the empirical potential needs to be evaluated in addition to the GAP. Nevertheless, these additional computational costs remain small compared to those needed to evaluate model A.

We can also use the above learning of differences to improve the prediction of the localised GP regression by explicitly accounting for long-range interactions. To this end, the empirical potential is chosen as the long-range Coulomb potential.

4. Descriptors of Local Atomic Environments

The success of nearly all methods in computational chemistry hinges on the quality of the encoding of the chemical structure into numerical input variables. This representation, also called the descriptor, has to capture the identity of the molecule or environment in terms of composition and configuration. Hundreds of different descriptors have been proposed, most of them for use in quantative-structure relationship studies (QSR) [55]. Generally, they can be classified into three categories depending on the level of included prior knowledge [56]: integrated, coarsened and first-principles-like. Integrated descriptors make use of experimentally measurable physical characteristics known to correlate well with the desired properties, whereas coarsened descriptors only take structural features into account. Both are tailored to the specific needs of a chemical compound, like molecules or crystalline solids, and are hence of limited generality.

We will focus on descriptors built from first principles, i.e. based only on the nuclear charge and the cartesian coordinates of the particle system in question. The reason for this decision lies in the Hamiltonian H which is uniquely determined by these properties, and in turn so are the state and energy of the system via the Schroedinger equation $H\psi = E\psi$. Hence, one expects to retain the complete informative value by forgoing all integrated properties and consequently to allow for a greater transferability across chemical compound space. The challenge consists of incorporating physically observable invariances while not losing uniqueness.

In this chapter we review both the physical and computational requirements needed for a reliable descriptor. We present different first-principle-like descriptors in use for the representation of local atomic environments before defining a new local descriptor based on the global Coulomb matrix introduced by Rupp *et al.* [45].

4.1. Physical and Computational Requirements for Descriptors

The desired properties stated in this subsection are a summary filtered from the discussion found in the papers by von Lilienfeld *et al.* [56], and Bartók *et al.* [3].

The most straightforward requirements for a descriptor used for the prediction

of energy values stem from the corresponding properties of the Hamiltonian and can be motivated physically. First of all, one wants the mapping from chemical environment to numerical representation to be unique. Assigning the same descriptor to different configurations not only contradicts physical intuition but also introduces systematic errors possibly hindering a successful learning procedure. Secondly, as the total energy of a system is independent of its orientation in space, the descriptor has to account for translation and rotation invariance. Additionally, one would expect that symmetrically equivalent atoms or particle groups contribute equally to the energy through the descriptor. Last but not least, the descriptor must be invariant with respect to the indexing of the atoms.

The above requirements of invariance are crucial to obtain reliable descriptors for the generation of physically accurate potentials. We now turn our focus to computational requirements that facilitate the practical aspects of the learning process. As such the most important feature is of course its computational cost. The determination of the descriptor must not outweigh the prediction of the property of interest. This implies that the dimension should be as low as possible and favorably independent of the specific chemical configuration in order to avoid costly adjustments when comparing different structures, e.g. environments with different numbers of neighbours.

At best one thrives for a closed analytic form that is able to cope with the various ranges relevant to physical chemistry. In order to keep numerical errors as small as possible, one demands continuity of the descriptor with respect to small variations in inter-atomic distances or nuclear charge. For physical applications, differentiability with respect to both is important as well.

All in all, one hopes to achieve that the property of interest, in our case the energy of a particle system, is a smooth function of the descriptor, as this greatly facilitates the learning process. In practice, however, this is difficult to check and performance of the descriptor has to be assessed using reference data and reliable error estimation methods such as cross validation.

4.2. Overview over other Local Descriptors in Use

In this section we provide a brief overview over other local descriptors employed in the generation of potentials via Machine Learning methods.

When first introducing the GAP in the physical review letter [4] in 2010, Bartók *et al.* used a modified bispectrum as a descriptor of mono-species crystalline environments. They form a local atomic density from the neighbours,

$$\rho(r) = \delta(r) + \sum_j \delta(r - r_{ij}) f_{cut}(|r_{ij}|), \tag{4.1}$$

coupled with a cutoff function

$$f_{cut}(r) = \begin{cases} 1/2 + \cos(\pi r / r_{cut})/2 & \text{if } r < r_{cut} \\ 0 & \text{else} \end{cases}, \tag{4.2}$$

to limit the spatial scale of the interactions. Summation over the neighbours renders it invariant to permutation of the atomic indices. In order to attain rotational invariance, the atomic density is projected on the surface of the four-dimensional unit sphere, this way retaining all information, even radial, from the 3D spherical region inside the cutoff. Expansion in the 4D spherical harmonics, the so-called Wigner matrices, leads to (infinitely many) coefficients from which the rotationally invariant bispectrum can be calculated as the triple-correlation.

In practice, the spatial resolution of the bispectrum has to be truncated. Nevertheless, even when using only up to 42 bispectrum coefficients, this descriptor is able to distinguish between different crystalline structures with a high accuracy. In principle, it can be extended to any multi-species atomic environment. For a more detailed description, refer to the supplementary information of the physical review letter cited above or to A. Bartók-Pártay's Ph.D. thesis [1].

Several different descriptors also featuring radial functions for the description of neighbourhood configurations at their core have been proposed. In [5], Behler discusses in detail a local descriptor of atomic environments using symmetry functions that explicitly incorporate the needed invariances. This descriptor was originally introduced by Behler and Parinello as input to high-dimensional neural networks [6]. Based on the same cut-off function defined in Equation (4.2) different radial and angular functions are proposed, such as

$$\begin{aligned} G^1 =& \sum_j f_{cut}(r_{ij}), \\ G^2 =& \sum_j e^{-\eta(r_{ij}-r_s)^2} \cdot f_{cut}(r_{ij}), \\ G^3 =& \sum_j \cos(\kappa r_{ij}) \cdot f_{cut}(r_{ij}), \\ G^4 =& 2^{1-\zeta} \sum_{j,k} (1 + \lambda \cos\theta_{ijk})^\zeta \cdot e^{-\eta(r_{ij}^2 + r_{ik}^2 + r_{jk}^2)} \\ & \cdot f_{cut}(r_{ij}) \cdot f_{cut}(r_{ik}) \cdot f_{cut}(r_{jk}) \end{aligned} \tag{4.3}$$

where η, r_s, κ, ζ, and λ are parameters and θ_{ijk} denotes the angle centered at atom i. A set of these functions with different parameter settings is then used to describe the distribution of neighbours within the cutoff sphere.

As with the bispectrum employed by Bartók *et al.*, the dimension of this descriptor has to be chosen empirically. Consequently, the uniqueness of this descriptor is impaired. In practice, the dimensions are chosen clearly larger than the number of of degrees of freedom, minimising the risk of introducing systematic errors via contradictory training data. For well chosen parameters, this leads to promising prediction results with a high accuracy. However, the

performance is very sensitive with respect to the chosen set of symmetry functions and the underlying cut-off radius. An inappropriate combination leads to a poor prediction. This is a downside of this descriptor, as there is little physical motivation for a particular combination and hence suitable sets have to be selected via an extra evaluation procedure.

A global molecular descriptor composed of local atomic contributions also using radial distribution functions is introduced by von Lilienfeld *et al.* [56]. They represent each atom in the molecule by its nuclear charge multiplied with a cosine term that has the radial distribution of all other atoms as an argument, effectively describing the atomic neighbourhood for an infinite cut-off radius. By summing over all atomic contributions, they obtain a Fourier series, fulfilling uniqueness and invariance requirements. The specific choice of radial distribution function is arbitrary. When employing a Gaussian function, the descriptor reads

$$FGR(r) = \sum_j Z_j^\alpha \cos(\frac{1}{Z_j} \sum_i Z_i \exp(-(r - r_{ij})^2/\sigma)) \tag{4.4}$$

where α and σ are hyperparameters. Von Lilienfeld *et al.* subject this version of the descriptor to preliminary testing on the same data set as was used for the evaluation of the Coulomb matrix in [45], a subset of the GDB-13 database. Combined with Gaussian kernel ridge regression as the Machine Learning method used for prediction, the Fourier series of Gaussian radial distribution functions (FGR) shows a predictive power on par with the Coulomb matrix.

The faithfulness of descriptors like the bispectrum employed by Bartók or the symmetry functions used by Behler can only difficultly be assessed using theoretical means because of the complicated algebraic dependency relationships between descriptor elements, making it unclear whether the number of independent degrees of freedom in the neighbourhood configuration, is (over-)achieved in terms of algebraically independent elements, i.e. if it is (over-)complete. For this reason, Bartók *et al.* evaluate it numerically by trying to reconstruct a reference configuration after perturbation of the atomic coordinates [3]. They find that for a fixed number of descriptor elements, the faithfulness decreases as the number of neighbours increases. However, the reconstruction quality improves with increasing descriptor length, in accordance with the expectation that the infinite series of the basis set expansion leads to overcomplete descriptors when not truncated. This suggests that the accuracy and completeness of these descriptors can be refined at will by including the truncation parameter into the set of hyperparameters.

In the same publication, Bartók *et al.* refine their bispectrum descriptor used for the GAP. Instead of considering only the representation of the atomic environment, they extend their design process to the (dis-)similarity measure used to compare the neighbourhoods, combining descriptor and kernel of the Gaussian process. Starting with directly defining the similarity of two atomic environments as the inner product of two neighbour densities, they obtain a

rotationally and permutationally invariant similarity kernel by integrating over all possible rotations $\hat{R}$ in three-dimensional space of one of the environments,

$$k(\rho, \rho') = \int \left| \int \rho(\mathbf{r}) \rho'(\hat{R}\mathbf{r}) \mathrm{d}\mathbf{r} \right|^3 \mathrm{d}\hat{R}. \tag{4.5}$$

In order to facilitate the evaluation of the angular integral, they change their construction of the atomic neighbour density from using Dirac-delta functions to using smooth Gaussians, expanded in terms of spherical harmonics Y_{lm} and radial basis functions g_n,

$$\rho(\mathbf{r}) = \sum_i \exp(-\alpha |\mathbf{r} - \mathbf{r}_i|^2) = \sum_{nlm} c_{nlm} g_n(r) Y_{lm}(\hat{\mathbf{r}}). \tag{4.6}$$

This allows the kernel to be calculated as the dot product of the bispectrum (see the original publication for details). Normalisation then leads to the general form of their SOAP (Smooth Overlap of Atomic Positions) kernel. They compare this combination to the standard bispectrum and Behler's descriptor used with the Gaussian kernel on silicon clusters and find an improvement both in reconstruction quality and in predictive power.

While this derivation of a similarity measure ultimately leads back to the bispectrum based on a modified atomic density and combined with a different kernel as before, its main difference lies in the elimination of many of the ad hoc choices necessary for descriptors and kernels. Although this ansatz has its advantages, we will continue to differentiate between the descriptor as numerical input and the kernel as similarity measure, as it allows for a more generalised framework.

4.3. Designing a Local Descriptor Based on the Coulomb Matrix

All of the local descriptors presented in the previous subsection show promising results for the generation of potentials when used in combination with Machine Learning methods. They have in common that they are all based on some kind of radial function for the description of atomic neighbourhoods. Some of them, like the bispectrum, require a computational cost not to be underestimated that can only be limited by restricting their resolution. A different and much simpler first-principles-like descriptor recently introduced with comparable prediction accuracy is the Coulomb matrix of Rupp *et al.* [45]. It is a molecular global descriptor in matrix form considering Coulomb interactions between atom pairs as off-diagonal entries and a polynomial fit of the nuclear charge to free atomic energies on the diagonal. The formal definition of its entries is given as

$$M_{ij} = \begin{cases} 0.5 Z_i^{2.4} & i = j, \\ \frac{Z_i Z_j}{||\mathbf{R}_i - \mathbf{R}_j||_2} & i \neq j, \end{cases} \tag{4.7}$$

where Z_i denotes the nuclear charge of the i-th atom, $\mathbf{R}_i$ its cartesian coordinates and the standard euclidean norm is used as a distance measure.

By construction, the Coulomb matrix is translation and rotation invariant, as it takes only inter-atomic distances into account. It is also continuous and differentiable with respect to nuclear charges and cartesian coordinates. In order to attain invariance with respect to atom-indexing, however, further measures have to be employed, since permutating the order of the atoms results in different matrices that can all be associated with the same molecule. In [32], Montavon *et al.* propose three methods to do so. The first is to not use the $n \times n$ Coulomb matrix itself as input to the learning method but its spectrum, the n sorted Eigenvalues. This solves the permutation problem but the sharp dimensionality reduction leads to a violation of the uniqueness criterion.

Their second proposition consists in selecting a representing permutation for the molecule out of all possibilities. They decide on sorting both rows and columns decreasingly according to the row norm to maintain symmetry. This approach does not violate the uniqueness of the descriptor and yields the desired effect.

Lastly, they suggest a sort of data set extension with the goal of dealing with the larger dimensionality of the Coulomb matrix compared to its spectrum. Instead of selecting just one permutated matrix as a representation, they draw several randomly sorted Coulomb matrices according to a conditional distribution afflicted with noise over all matrices associated with one molecule.

When comparing the performance of the three variants, Montavon *et al.* find that the random Coulomb matrices perform slightly better than the sorted variant at the cost of considerably larger training set sizes.

Our goal is now to design a local descriptor capitalising on the simplicity of the Coulomb matrix while still achieving an accuracy comparable to the descriptors employed by Bartók for the GAP in [4] and later publications.

4.3.1. The Localised Coulomb Matrix

The Coulomb matrix is a global descriptor of a molecule. As such it can only be applied to a finite number of particles, making it a priori non-applicable to infinitely periodic crystals. In our case, the decomposition of the total energy as a sum of atomic contributions depending only on the local environment within a suitable cut-off radius allows us to use a molecular descriptor such as the Coulomb matrix even for infinite structures. In order to apply it to the individual environment of a specific atom, however, it needs to be related to the atom in question in some way, resulting in a local instead of global representation. We propose to combine the entries of the Coulomb matrix with the SNCF metric (also called the British Rail metric or Post Office metric), effectively scaling the contribution of each atom pair by its distance to the atom in the center.

For a fixed central atom, consider its neighbours located within a given cut-off

radius. Following the example of the Coulomb matrix, we construct a matrix with entries for each possible particle pair, resulting in the matrix dimension of number of neighbours plus one (the central particle itself) squared. An upper bound n_{max} can be calculated for the number of neighbours for any fixed radius using the maximum packing efficiency for crystalline solids or similar concepts extended to general particle systems. Hence by introducing dummy atoms when necessary, the dimension of the descriptor can be rendered independent of the actual number of neighbours currently residing in the neighbourhood.

For the diagonal entry describing the contribution of the central atom itself we adopt the polynomial fit of the nuclear charge to free atomic energies used in the definition of the diagonal entries of the Coulomb matrix. In all other entries, we adjust the denominator of the Coulomb interaction term by interchanging the inter-atomic distance between the two atoms with the metric induced by the euclidean norm to the SNCF metric,

$$d_c(\mathbf{R}_i, \mathbf{R}_j) = ||\mathbf{R}_i - \mathbf{R}_c||_2 + ||\mathbf{R}_j - \mathbf{R}_c||_2, \tag{4.8}$$

where $\mathbf{R}_c$ refers to the cartesian coordinates of the central atom, and $\mathbf{R}_i$, $\mathbf{R}_j$ to the respective coordinates of the particle pair considered, one of which can well be the central atom itself. We call the resulting matrix *localised Coulomb matrix* in order to emphasise its origin.

A p-particle system is then described by p localised Coulomb matrices of dimension $n_{max} + 1 \times n_{max} + 1$, each corresponding to the environment of one atom in the system, defined for the k-th atom as

$$M(k)_{ij} = \begin{cases} 0.5 Z_i^{2.4} & i = j = k, \\ \frac{Z_i Z_j}{||\mathbf{R}_i - \mathbf{R}_k||_2 + ||\mathbf{R}_j - \mathbf{R}_k||_2} & \text{otherwise.} \end{cases} \tag{4.9}$$

Just as the Coulomb matrix, this descriptor is translation and rotation invariant by construction, as well as continuous and differentiable with respect to the inter-atomic distances and nuclear charges. We achieve permutation invariance concerning atom-indexing by following Montavon *et al.*'s second proposition. While fixing the row and column belonging to the central atom as the first row and column respectively in each local matrix, the other rows and columns are sorted with respect to their row norm.[1]

We want to stress that this proposition of a localised Coulomb matrix is by far not the only possible choice. For example it is not clear, whether the above definition of diagonal entries not corresponding to the central atom is the most suitable for application purposes. By weighting the nuclear charge of the atom with the inverse distance to the central atom, an exponential decay on the diagonal is introduced, which is in accordance with the localisation assumption. It is however possible, that this decay masks the unique identity of the atom, e.g. that there is no difference in the diagonal entries of a small hydrogen atom

[1] **Revised Version:** Here the incorrect statement about the uniqueness of the descriptor was removed.

near the central atom and a carbon atom further away. In order to examine this effect, we introduce a second definition of the localised Coulomb matrix, in which the diagonal is not weighted. We refer to this variant as *localised Coulomb Matrix WLD* (weightless diagonal). A p-particle system is then as before described by p matrices, now defined for the k-th atom as

$$M(k)_{ij}^{WLD} = \begin{cases} 0.5Z_i^{2.4} & i = j, \\ \frac{Z_i Z_j}{||\mathbf{R}_i - \mathbf{R}_k||_2 + ||\mathbf{R}_j - \mathbf{R}_k||_2} & \text{otherwise.} \end{cases} \tag{4.10}$$

Additionally, one could of course deviate stronger from the original Coulomb matrix in choice of underlying norm or manner of incorporating the type of atom. The Coulomb matrix has, however, proven its performance at least for biomolecular applications, so we refrain from such fundamental modifications.

We expect that the localisation extends the suitability of this descriptor from biomolecules to also include crystalline data sets. Using a global ansatz, this was not possible until now, even though attempts have been made. In [48], Schütt *et al.* propose two global descriptors for crystal structures based on the Coulomb matrix by applying it either to the Bravais matrix conventionally used in the solid state community or to the k nearest neighbours of a fixed atom. However, they find that both descriptors are outperformed by a third variant not related to the Coulomb matrix but built using partial radial distribution functions.

In order to verify our expectation, we perform tests of this descriptor on biomolecular data as well as on silicon crystal structures. For the results refer to Chapter 6.

4.3.2. Examining the (Non-)Uniqueness of the Localised Coulomb Matrix

As the uniqueness of the mapping from environment to numerical representation is crucial for a successful inference of physical properties using interpolation of the PES, we now examine it for our localised Coulomb matrix descriptor in detail.[2]

Consider the environment of a particle P with an arbitrary number of neighbours $N_1, ..., N_{p^*}$. If $p^* < n_{max}$, dummy atoms with nuclear charge zero are added. We assume w.l.o.g that the neighbours are already indexed in such an order that we obtain the matrix whose rows are decreasingly ordered according to their row norm, i.e. that no further permutation is necessary. Then the localised Coulomb matrix used to describe this neighbourhood has the following

[2] **Revised Version:** This whole subsection has been rewritten to address the inability of the localised Coulomb matrix to distinguish between atomic environments differing only in local rotations of the atoms around the central atom (but not in atom types or distances to the central atom).

entries

$$M(P)_{ij} = \begin{cases} 0.5 Z_P^{2.4} & i = j = 0, \\ \frac{Z_P Z_{N_j}}{||\mathbf{R}_{N_j} - \mathbf{R}_P||_2} & i = 0 \text{ and } j \neq 0, \\ \frac{Z_P Z_{N_i}}{||\mathbf{R}_{N_i} - \mathbf{R}_P||_2} & j = 0 \text{ and } i \neq 0, \\ \frac{Z_{N_i} Z_{N_i}}{2||\mathbf{R}_{N_i} - \mathbf{R}_P||_2} & i = j \text{ and } i \neq 0, \\ \frac{Z_{N_i} Z_{N_j}}{||\mathbf{R}_{N_i} - \mathbf{R}_P||_2 + ||\mathbf{R}_{N_j} - \mathbf{R}_P||_2} & \text{otherwise,} \end{cases} \tag{4.11}$$

where we have explicitly listed the cases of the first row and column, when one of the two particles considered corresponds to P.

If this descriptor is not unique, we can find an environment of a different particle P' with the same number of neighbours $N'_1, ..., N'_{p^*}$ but with a different chemical configuration whose associated localised Coulomb matrix is identical with $M(P)$. This means that there exists a permutation $\pi : \{0, ..., p^*\} \to \{0, ..., p^*\}$ with $\pi(0) = 0$ such that

$$M(P)_{ij} = M(P')_{\pi(i)\pi(j)} \text{ for all } i, j = 0, ..., p^*. \tag{4.12}$$

As we are fixing the row and column belonging to the central particle to be the first in every matrix, we immediately obtain that both particles P and P' have to be of the same type, i.e. $Z_P = Z'_P$. Furthermore, using the other entries in the first row and the diagonal entries, we obtain the following two equivalences,

$$\begin{aligned} \frac{Z_{N_j}}{||\mathbf{R}_{N_j} - \mathbf{R}_P||_2} &= \frac{Z_{N'_{\pi(j)}}}{||\mathbf{R}_{N'_{\pi(j)}} - \mathbf{R}_P||_2}, \\ \frac{Z_{N_j}^2}{||\mathbf{R}_{N_j} - \mathbf{R}_P||_2} &= \frac{Z_{N'_{\pi(j)}}^2}{||\mathbf{R}_{N'_{\pi(j)}} - \mathbf{R}_P||_2}, \end{aligned} \tag{4.13}$$

from which we can deduce that

$$Z_{N_j} = Z_{N'_{\pi(j)}} \text{ and } ||\mathbf{R}_{N_j} - \mathbf{R}_P||_2 = ||\mathbf{R}_{N'_{\pi(j)}} - \mathbf{R}_P||_2. \tag{4.14}$$

Hence, the composition of the environments in terms of nuclear charges of the neighbours and inter-atomic distances with respect to the central atom have to coincide. However, this does not uniquely determine the atomic environment as it provides no angular information about the position of two different neighbours relative to each other. This means that the definition of the localised Coulomb matrix as provided in Equation (4.9) is not able to distinguish between environments differing only in local rotations of the neighbours around the central atom. Two such environments resulting in identical localised Coulomb matrices are depicted exemplarily in Figure 4.1.

The reason for this "angular blindness" of the localised Coulomb matrix lies in the way the localisation was achieved by measuring all distances (only) with respect to the central atom. Consequently, in order to render the localised

Figure 4.1.: Two example atomic environments resulting in identical localised Coulomb matrices as they differ only with respect to the angles between the neighbours.

Coulomb matrix truly unique for any atomic environment we would need to provide the missing angular information by including also the distances between any two atoms in the environment, as it is the case for the global Coulomb matrix as given by Equation (4.7). A possible redefinition would therefore be

$$M(k)_{ij} = \begin{cases} 0.5 Z_i^{2.4} & i = j = k, \\ \frac{Z_i Z_j}{||\mathbf{R}_i - \mathbf{R}_k||_2 + ||\mathbf{R}_j - \mathbf{R}_k||_2 + ||\mathbf{R}_i - \mathbf{R}_j||_2} & \text{otherwise}, \end{cases} \tag{4.15}$$

where the contribution of an atom pair is now also weighted by the inverse of their direct distance, thereby fixing the angle between them and the central atom.

We want to stress that this deficiency in the uniqueness of the localised Coulomb matrix is of course only an issue for the efficient approximation of the potential energy surface, if the provided training and test data actually consist of environments which differ only in the angles between the neighbours. When describing the data used for evaluation in Section 5.1, we will show that neither the biomolecular nor the silicon data set exhibit the pathological case, meaning that the environments present in these data sets are uniquely represented by the localised Coulomb matrix as defined in Equation (4.9). Hence, all results obtained in this thesis can be expected to extend directly to a unique version.

4.3.3. Reinforcing the Localisation Effect

In molecular dynamics applications, particles enter and leave the atomic vicinity dynamically in between timesteps as part of the simulation of nanoscale processes. This behaviour introduces a sort of noise into the representation of nearly identical neighbourhoods. In order to limit the influence of this noise, the descriptor should have a smooth decay at the borders of the individual atomic environment. This can be achieved by different approaches, e.g. by multiplication of a smooth cutoff function. For the localised Coulomb matrix we choose to reinforce the penalisation of the distance of contributing particles to the central atom, leading to a stronger decay of the entries and a reduction of noise. This is done by introducing a parameter α as the exponent of the denominator of the entries and setting it to values larger than one, e.g. inspired

by the Lennard-Jones potential, $\alpha = 6$. The adaptation is then defined as

$$M(k)_{ij} = \begin{cases} 0.5Z_i^{2.4} & i = j = k, \\ \frac{Z_i Z_j}{(||\mathbf{R}_i - \mathbf{R}_k||_2 + ||\mathbf{R}_j - \mathbf{R}_k||_2)^\alpha} & \text{otherwise.} \end{cases} \tag{4.16}$$

This stronger penalisation of the distance effectively reduces the size of the region, in which particles contribute significantly, resulting in a higher localisation of the descriptor. We expect localised Coulomb matrices with this raised exponent to be particularly well-suited for molecules without dipoles where all interactions between atoms can be assumed to be short-range only. We will however verify its performance unbiasedly also on regular biomolecules and crystalline solids.

4.3.4. Derivatives of the Localised Coulomb Matrix

As described in Section 3.3, the derivatives of the descriptor with respect to the cartesian coordinates of the particle system are needed, when predicting the gradient or including gradient information into the training data to enhance function value prediction. Due to its simple structure, this is not a problem with the localised Coulomb Matrix. We state here the derivative of the standard variant.

Consider a given system consisting of P particles with cartesian coordinates $\mathbf{x} \in \mathbb{R}^{P \times d}$. Then we can write the descriptor q_k of the k-th particle built using the localised Coulomb matrix formally as

$$q_k^{LCM} = F \Pi M(k) \Pi, \tag{4.17}$$

where Π denotes the permutation matrix that takes care of the correct sorting and F the storage operator that flattens the matrix into a vector row-wise. Both Π and F are independent of the cartesian coordinates of the system, hence we have for the derivative of q_k with respect to the cartesian coordinate x_p^t, $t = 1, ..., d$, of the p-th particle,

$$\frac{\partial q_k}{\partial x_p^t} = F \Pi \left(\frac{\partial M(k)}{\partial x_p^t}\right) \Pi, \tag{4.18}$$

meaning that the differentiation only affects the entries of $M(k)$. For those it holds,

$$\frac{\partial M(k)_{ij}}{\partial x_p^t} = \begin{cases} \frac{\alpha Z_i Z_j}{(||\mathbf{R}_i - \mathbf{R}_k||_2 + ||\mathbf{R}_j - \mathbf{R}_k||_2)^{\alpha+1}} \left(\frac{x_i^t - x_k^t}{||\mathbf{R}_i - \mathbf{R}_k||_2} + \frac{x_j^t - x_k^t}{||\mathbf{R}_i - \mathbf{R}_k||_2} \right), \\ \qquad\qquad \text{if } p = k \text{ and } (i \neq k \text{ or } j \neq k) \\ \frac{-\alpha Z_i Z_j (x_p^t - x_k^t)}{(||\mathbf{R}_i - \mathbf{R}_k||_2 + ||\mathbf{R}_j - \mathbf{R}_k||_2)^{\alpha+1} ||\mathbf{R}_p - \mathbf{R}_k||_2}, \\ \qquad\qquad \text{if } (p = i \text{ and } i \neq k) \text{ or } (p = j \text{ and } j \neq k) \\ 0 \qquad\qquad \text{else} \end{cases} \tag{4.19}$$

where as before $\mathbf{R}_i = (x_i^t)_{t=1}^d$.

5. Assessment and Validation

In this chapter we specify all details concerning the validation of accurately interpolating the PES via the LC-GAP. For the results of this evaluation approach refer to Chapter 6.

Starting with the description of the employed data sets and any preparatory steps executed on them, we provide the example implementation of the localised Coulomb matrix in pseudo-code and analyse its computational cost. Next, we present the library used for the generation of the potentials and derive the computational complexity of the framework.

Finally, we depict the cross validation procedure used for the assessment of the predictive power, as well as how the hyperparameters are chosen using both nested cross validation and maximisation of the likelihood.

5.1. Data Sets

The concept of localised GP regression in combination with localised Coulomb matrices as a descriptor of local atomic environments is validated on two different data sets, demonstrating the versatility of the ansatz.

5.1.1. The Biomolecular Data Sets

The data set QM7 used in [45] and [20] for the validation of the Coulomb matrix as a suitable global molecular descriptor is a subset of the GDB-13, a database enumerating nearly a billion small druglike organic molecules [9]. QM7 itself consists of 7165 biomolecules composed of up to 7 heavy atoms (C, N, O, S) and saturated with hydrogen. It is freely available, both in a Matlab and in a Python readable format [40]. Aside from the precalculated (global) Coulomb matrices and the target atomisation energies, splits for executing a cross validation procedure with five runs are also provided, as well as the nuclear charges and cartesian coordinates of the composing atoms. Hence all information needed for the calculation of localised Coulomb matrices is provided, making it the ideal choice for comparing the predictive power of this local descriptor to that of the global one it is based on.

As stated in [45], the QM7 data set includes constitutional isomers which consist of the same atoms but differ in the connecting bonds, but it does not contain conformational isomers, meaning that no two chemical compounds from the

data set can be interconverted by rotation about single bonds. Hence, the pathological case where the localised Coulomb matrix is not able to uniquely represent the atomic environments (cf. Section 4.3.2) is explicitly excluded and this data set is well-suited for the evaluation of this descriptor.[1]

In order to obtain multiple data sets with a natural ordering concerning their size, QM7 is filtered into the (non-disjoint) subsets QM4, QM5 and QM6, each comprised of molecules with up to 4, 5 and 6 heavy atoms respectively. These three data sets consisting of 59, 217 and 1167 molecules, are mainly used for validating the framework due to their manageable sizes. For this, two further preparations steps are necessary. In the first step, the organisation of the chemical configurations is restructured from individual arrays for coordinates, nuclear charges and atomisation energies to a list of `AtomSystem`s, in which each molecule is represented as its own instance with a number of atoms n, an array of coordinates of dimension $n \times 3$, an array of nuclear charges of dimension n and an associated atomisation energy value. In a second step, the representations of the local atomic environments are calculated once for each variant of the localised Coulomb matrix and value of the cut-off radius as described in the following section. They are stored as another property of the respective `AtomSystem`, resulting in multiple copies of the data sets only differing in the specific descriptor chosen. On these data sets, the cross validation is carried out, individually assessing the performance of the different variants.

There are two fundamentally different settings for the cut-off radius on these data sets: none, or a finite value in $\mathring{A}$. The first option corresponds to atomic environments that take all other atoms in the molecule into account, independent of their distance to the central atom. This means the atomic decomposition described in Section 3.2 makes no assumption at all about the atomic contributions only depending on a local neighbourhood. We start with this option in our analysis in Chapter 6 as it is well-suited for comparing the localised Coulomb matrix with its global basis. It stands in contrast to the second option, where one restricts the influence of the neighbours to the atomic energy to include only those within a finite region.

Depending on the option, the matrix dimensions are calculated differently. With a finite cut-off radius, the maximum number of neighbours is determined via a dense packing of methane molecules, (cf. Subsection 5.2.1 for details). With the cut-off radius set to none, all environments have the dimensions of the largest molecule present in the data set. This is necessary as all environments are required to have the exact same dimensions for comparison within the learning scheme. This means, however, that when one aims to test a potential trained e.g. on QM5 on a data set containing larger molecules, e.g. QM7, the dimensions of the localised Coulomb matrix of the smaller data set have to be adjusted by padding with zeros. This extra preparatory step is done for the validation of the transferability of potentials built using the localised Coulomb matrix, the

[1] **Revised Version:** This paragraph was added in order to examine whether the atomic environments present in this data set are uniquely represented by the localised Coulomb matrix.

Data set	Excluded Configuration(s)
QM5_onlyC*	1,3-pentadiyne
QM6_onlyC*	1,3-pentadiyne, benzene
QM7_onlyC*	1,3-pentadiyne, benzene, toluene, 1,3,5-heptatriyne

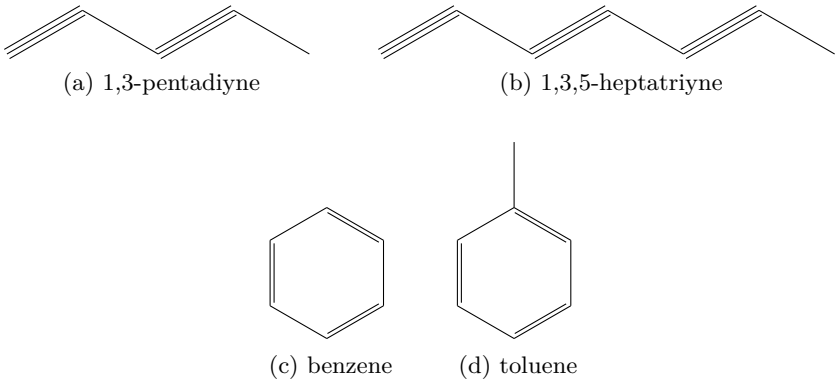
Table 5.1.: Systems excluded from the data sets QMx_onlyC* compared to QMx_onlyC

results of which are presented in Subsection 6.1.4.

Data Sets Featuring only Carbon and Hydrogen Atoms

As described in Subsection 4.3.3, we would like to test the influence of an exponent larger than one in the denominator of the entries of the local Coulomb matrix on its predictive power. Since a higher exponent reinforces the localisation of the descriptor, we expect the performance to improve especially on data sets featuring only short-range interactions. To this end, the data sets QM4, QM5, QM6 and QM7 are filtered once again, this time as to contain only molecules consisting of carbon and hydrogen, effectively banning all possible dipoles. The corresponding sets are labeled with the suffix *_onlyC*. They consist of 20, 49, 151 and 498 molecules respectively. The data set QM4 is not considered in the evaluation process due to its small training set size.

The data preparation steps concerning restructuring of the data organisation and calculation of the atomic environment representations are executed as described above. However, upon inspection of the regression results for these filtered data sets, it becomes apparent that a further data processing step is necessary. The drastic reduction of the number of molecules present in these data sets leads to the problem of strong outliers on which prediction of the atomisation energies fails significantly when compared to the mean prediction error. This is for example the case for benzene in QM6_onlyC, the only aromatic molecule in the data set. With all interpolation techniques, the prediction is only valid as long as one remains in the range of the interpolation points. In our case, the quality of the generated potential depends strongly on the learned systems. For a chemical configuration that differs significantly from those present in the training data, prediction is bound to fail. For our purposes we decide to exclude all systems whose absolute difference in prediction deviates more than eight times the standard deviation from the mean absolute error. The data sets are modified into the sets QM5_onlyC*, QM6_onlyC* and QM7_onlyC* as summarised in Table 5.1, where the indicated molecules feature unique configurations and hence do not allow for inference from the other training molecules. Figure 5.1 depicts the structural formula for the excluded molecules.



(a) 1,3-pentadiyne (b) 1,3,5-heptatriyne

(c) benzene (d) toluene

Figure 5.1.: Skeletal formula of the molecules excluded in QMx_onlyC*. At each vertex a carbon atom is located. Additionally, each carbon atom is understood to be saturated with hydrogen atoms such that each carbon atom features four bonds.

5.1.2. The Silicon Data Sets

The biomolecular data sets QMx are suited to assess whether the localised Coulomb matrix can be employed successfully when learning across chemical compound space, i.e. for the prediction of energies of different chemical compounds, all at equilibrium configuration. However, a second kind of application requires the PES of just one chemical compound to be interpolated. The difficulty here lies in accurately predicting not only the local minima which represent the chemically meta-stable states, but also structures deviating from the equilibrium configuration, e.g. in reaction path finding applications.

For this case we prepare a second kind of data sets, based on the semiconductor silicon, a crystalline solid. Also, this allows us to verify the capability of the localised Coulomb matrix to cope with periodic infinite chemical structures. Additionally, the calculation of forces is easier on crystals than on molecules as there are less degrees of freedom to perturb. Hence, we will use these data sets to evaluate the prediction of gradient values within the localised GP regression framework.

The generation of these crystalline data sets is done using the *Atomistic ToolKit (ATK)* [41]. This software package provides an interface to atomic-scale modelling using DFT methods, refer to [10] and [51]. As a basis for the data sets the silicon 8-atom supercell is chosen, as shown in Figure 5.2. Silicon crystallises in a diamond structure, with a lattice constant of $5.4306\mathring{A}$ in equilibrium configuration.

Three different data sets are constructed. First, only the cartesian coordinates of the atoms are randomly perturbed by up to $p_A\mathring{A}$, leading to the data set Si8ApCC. In Si8ApLV, perturbation of up to $p_{LV}\mathring{A}$ only affects the coordinates of the lattice vectors. Lastly, both the lattice vectors and the atoms are per-

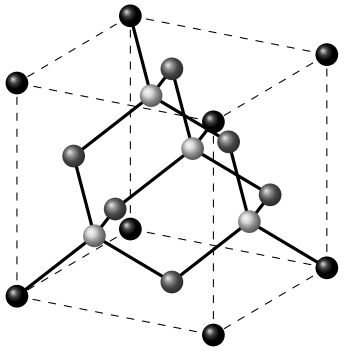


Figure 5.2.: 8-Atom supercell in diamond structure. The diamond structure is obtained by placing the primitive cell composed of two tetrahedrally bonded atoms at the positions of the face-centered cubic Bravais lattice. When counting, atoms are weighted by the inverse number of supercells they belong to ($8*\frac{1}{8}+6*\frac{1}{2}+4=8$). Shading is added for better visuality (corners: black, faces: grey, interior points: light grey).

turbed simultaneously in the data set Si8ApALV. The perturbation values are drawn randomly from a uniform distribution on the interval [-p_A, p_A] or [-p_{LV}, p_{LV}] respectively. All three data sets describe a neighbourhood of the equilibrium configuration in the high-dimensional Born-Oppenheimer PES. However they differ in the number of degrees of freedom which are included in the perturbation. In Si8ApCC and Si8ApLV only subspaces of the PES are sampled.

Due to the construction of these data sets via random perturbation of a basis cell, it is highly unlikely that atomic environments will be generated where the neighbours are situated at the exact same distance from the central atom, but differ in the angle. The possible decline in performance of the learning method due to non-unique neighbourhood representations is hence negligible and we deem these data sets suitable for the evaluation of the LC-GAP.[2]

We stress that also due to the construction procedure of the data sets, the 8-atom basis supercell itself is not necessarily included. This is important to keep in mind for the analysis of the results obtained in Section 6.2. The results are bound to improve by enforcing the inclusion of the basis supercell, which we note as an important enhancement for the future.

Each perturbed supercell configuration is stored as a ATK `BulkConfiguration`. Calculation of the total energies and forces is done using the numerical orbital model implemented in the an ATK-DFT `LCAOCalculator` with the following settings:

- Basis set: DoubleZetaPolarized,
- Spin: polarised,
- Exchange correlation: SGGA with PBES,

[2] **Revised Version:** This paragraph was added in order to examine whether the atomic environments present in this data set are uniquely represented by the localised Coulomb matrix.

- K-Point sampling: (5,5,5),
- Electron temperature: 1000K.

For the mathematical basics concerning density functional theory we refer to the books [46] and [36] and the references therein.

As with the biomolecular data sets, the descriptors are calculated once per configuration and parameter settings, and stored for future access. Thus, the data set is ready for the evaluation of the LC-GAP. During a validation run a fixed number of configuration files storing the bulk configuration along with energy and forces is randomly selected from the data repository and the required descriptors read from associated files.

5.2. Calculation of the Localised Coulomb Matrix

In this section we describe the calculation of the localised Coulomb matrix. For our validation of this descriptor when used in combination with the GAP framework, we use a Python implementation. It is based on an abstract class called `LocalEnvironmentCalculator` defining a method `calculcate_local_environments`. Parameters for this method are the particle system itself, the localisation exponent α, the cut-off radius and the lattice type. The different variants of the localised Coulomb matrix inherit from the `LocalEnvironmentCalculator` class and specify the environment calculation according to their definition provided in Section 4.3.

In Listing 5.1 a Python-like pseudo-code description of the standard variant of the localised Coulomb matrix is shown. Of course, the implementation has to be adjusted to the particular requirements of the `AtomSystem` or `BulkConfiguration` class, respectively.

The procedure for calculating the numerical representation of the atomic environments of a given system is as follows. First, the maximum number of particles possible within a given cut-off radius is calculated (refer to the subsequent subsection for the details). The dimension of the descriptor is then fixed using this value. When exploiting the symmetry of the localised Coulomb matrix only the upper triangular part needs to be stored.

Next, we iterate over every atom in the given system. Its actual neighbours are determined and the localised Coulomb matrix entries filled by iterating over their list in two nested for-loops, again making use of its symmetry property. Invariance with respect to atom-indexing is ensured by sorting the matrix both row- and column-wise decreasingly according to the row-norm. The row and column of the central particle, however, remain fixed as the first row and column of the matrix, thereby conserving uniqueness of the descriptor. The individual matrix (respective its upper triangular part) is flattened and stored row-wise as a vector.

Lastly, the environments for the whole system are combined into one large

matrix, where each row corresponds to one atom, and returned.

When predicting forces, the `calculate_local_environments` method is extended to include the calculation of the gradient of the localised Coulomb matrix (cf. Section 4.3.4). We do not describe it here in more detail, as the procedure is similar to the assembly of the localised Coulomb matrix itself. Note that the same sorting permutation has to be applied in order to ensure correct indexing.

When handling very large particle systems it may be more useful to calculate the environment of each atom individually at prediction time in order to reduce storage costs, instead of assembling all environments for a given system together.

5.2.1. Calculation of the Matrix Dimensions for a Given Cut-off Radius

In order for the localised Coulomb matrix to be useful as a descriptor of atomic environments in practice, its dimension has to depend merely on the fixed cut-off radius. This can be done using a dense packing of spheres with a general packing factor and a pessimistic assumption about the smallest covalent atom radius possible. This way the dimensions of the descriptor are independent of the actual number of neighbours residing in the particular atomic environment.

For most chemical compounds, however, this will largely exceed the actual storage capacity required. Since the training data can be expected to be more or less homogeneous with respect to its chemical nature (it does not make sense to predict crystalline structures using a Gaussian approximation potential trained on organic molecules), we will relax this requirement slightly and allow the dimension to additionally depend on the basic chemical layout, such as the lattice type for crystallines.

For the two data sets we use for validation the dimensions are fixed in detail as follows. First of all, concerning the silicon crystalline data set, the calculation can be based in a straightforward manner on its lattice type and the corresponding atomic packing factor. Silicon crystallises in the diamond cubic lattice structure which has a density of $\frac{\pi\sqrt{3}}{16} \approx 0.34$. Its covalent radius is $r_{\text{silicon}} = 1.11\,\mathring{A}$. We add this value to the cut-off radius to ensure inclusion of atoms situated exactly at the cut-off border. This results in the following formula where the volume of the enlarged cut-off region multiplied with the atomic packing factor is divided by the approximate volume of the silicon atom,

$$N_{max} = \left\lceil \frac{\pi\sqrt{3}(r_c + r_{\text{silicon}})^3}{16(r_{\text{silicon}})^3} \right\rceil . \tag{5.1}$$

In contrast, the optimal formula is less obvious for the biomolecular data set QM7 and its subsets. Empirical testing led us to basing the maximum dimension of the descriptor on a dense packing of methane molecules with an effective radius of $r_{\text{methane}} = 1.4\,\mathring{A}$ and the densest atomic packing factor of $\frac{\pi}{3\sqrt{2}} \approx 0.74$.

```
calculate_local_environments(atom_system,
                             alpha,
                             cut_off_radius,
                             lattice_type):

    n_particles_max = get_number_particles(cut_off_radius,
                                           lattice_type)

    if UT: # only store upper triangular part
        dim = n_particles_max*(n_particles_max+1)/2
    else:
        dim = n_particles_max*n_particles_max
    environments = zeros(atom_system.n_atoms, dim)

    # calculate local environment for each atom
    for atom in atom_system:
        neighbours_list = get_neighbours(atom,
                                         atom_system,
                                         cut_off_radius)
        M = zeros(len(neighbours_list)+1, len(neighbours_list)
            +1)
        M[0, 0] = 0.5*atom.nuclear_charge^^2.4

        # assume neighbours have indices from 1 to
        # n_particles_max-1 (or less)
        for neighbour_i in neighbours_list:
            M[0, neighbour_i.index] = atom.nuclear_charge*
                neighbour_i.nuclear_charge/norm(neighbour_i.
                coordinates-atom.coordinates)^^alpha
            # exploit symmetry
            M[neighbour_i.index, 0] = M[0, neighbour_i.index]

            for neighbour_j in
                    neighbours_list[neighbour_i.index:]:
                M[neighbour_i.index, neighbour_j.index] =
                    neighbour_i.nuclear_charge*neighbour_j.
                    nuclear_charge/(norm(neighbour_i.coordinates
                    -atom.coordinates)+norm(neighbour_j.
                    coordinates-atom.coordinates))^^alpha
                # exploit symmetry
                M[neighbour_j.index, neighbour_i.index] =
                    M[neighbour_i.index, neighbour_j.index]

        # take care of sorting (decreasing order)
        row_norms = sum(M[1:,:], axis=0)
        sorting_permutation = argsort(row_norms)[::-1]+1
        # adjust for central atom
        sorting_permutation = concatenate([0],
                                          sorting_permutation)
        M = M[:, sorting_permutation]
        M = M[sorting_permutation, :]

        if UT:
            M = M.upper_triangular_part()
        environment[atom.index] = M.flatten()
    return environments
```

Listing 5.1: Pseudo-code description of the implementation of the standard variant of the localised Coulomb matrix.

Additionally, we again adjust the cut-off radius r_c in order to surely include all atoms on the border, this time by the covalent radius $r_{\text{sulfur}} = 1.05\,\mathring{A}$ of sulfur, the biggest atom in the data set. Hence, in this case we have

$$N_{max} = \left\lceil \frac{\pi(r_c + r_{\text{sulfur}})^3}{3\sqrt{2}(r_{\text{methane}})^3} \right\rceil . \tag{5.2}$$

5.2.2. Computational Cost

We now consider the computational cost of the localised Coulomb matrix for a given particle system with P particles. Based on the Listing 5.1, we observe that for each particle we need to determine its neighbours and then iterate over them in a double loop in order to consider all possible pairs. The determination of its neighbours is done in the worst case by considering all other particles. However, using a suitable data structure such as the *linked cell* method [16], this can easily be improved. We therefore argue that the assembly of the localised Coulomb matrix is the more expensive operation.

The maximum number of neighbours is limited depending only on the cut-off value by `n_particles_max`. Hence, in order to fill the entries of the localised Coulomb matrix, we have a computational cost of $\mathcal{O}(\texttt{n\_particles\_max}^2)$, where the multiplicative constant is improved by exploiting its symmetry and the fact that in practice of course, iteration is done only over the actual number of neighbours of the particular atom.

Summing over all particles, we arrive at a cost of $\mathcal{O}(P * \texttt{n\_particles\_max}^2)$ for the calculation of all environments for a given system. Here, we assume that owing to our localisation assumption $\texttt{n\_particles\_max} \ll P$. Since `n_particles_max` does not actually depend on P, we conclude that the complexity required for the calculation of the descriptors of a P-particle system is $\mathcal{O}(P)$, i.e. linear in the number of particles in the given system.

5.3. Generation of the GAP

We use the `mgauss` library developed by the Virtual Materials group at Fraunhofer SCAI for the generation of the LC-GAP. It is written in C and provides a Python interface. It implements both the standard Gaussian process regression described in Section 2.1.1 and the localised variant presented in Section 3.2. Additionally, it provides optimisation routines for the model selection of the hyperparameters via maximisation of the marginal likelihood (cf. Section 5.4.2).

The Python interface is essentially determined by four classes:

`TrainingData` stores the training data by providing a method
`addLocalizedData(input, target)`

`Kernel` specifies the kernel used by the Gaussian process by providing different subclasses such as `IsotropicSquaredExponential(hyperparameters)`

`HyperparameterOptimizer` specifies the optimisation routines for the model selection by providing methods such as `setRelativeTolerance` and `setMaxIterations`.
If no optimisation is needed, set to None.

`GaussianApproximation` generates the learnt potential via (localised) Gaussian process regression by providing the method `generate(training_data, kernel, optimiser)`.
Prediction is then done using the method `evaluate(input)`.

The optimisation routines provided by the `HyperparameterOptimizer` class are based on the NLopt nonlinear-optimisation package [23]. We use the *Constrained Optimization by Linear Approximations* method, `NLOPT_LN_COBYLA`, as a default with a relative tolerance of 10^{-6} and a maximum number of iterations of 1000. Its mathematical theory can be found in the publication [38].

5.3.1. Computational Cost

We now analyse the computational cost of the Gaussian process regression based on atomic energy contributions, as summarised in Equation (3.10). As we will see, the complexity is not affected by introducing the localisation.

We perform our analysis in two steps, separating between learning and prediction.

Learning of the Potential

First, we consider the cost of the generation of the potential as a function of the number of training examples N. For the Gaussian process regression, learning consists of calculating the coefficients of the linear combination of kernel basis functions which determine the interpolation. They are given by

$$\alpha = L(L^T \mathbf{C}_K L + \sigma_\varepsilon^2 I)^{-1} \mathbf{y}. \tag{5.3}$$

From this equation we identify the following necessary steps:

1. Calculate the entries of the covariance matrix $\mathbf{C}_K$,
2. Multiply with the sparse matrix L and its transpose,
3. Add the perturbation,
4. Invert the perturbed covariance matrix ,
5. Multiply with observations $\mathbf{y}$,
6. Multiply again with sparse matrix L.

Here, K denotes the number of atomic environments counted over all training examples. Of course, the covariance matrix C_K of the atomic environments should never be assembled in practice but always be directly multiplied with the matrices L and its transpose.

Without loss of generality we assume that each training system features P atoms, hence K is a linear function of N given by $K = P * N$. We can use this to derive that the first step has a complexity of $\mathcal{O}(K^2) = \mathcal{O}(N^2)$, since the number P of particles can be considered as a constant. Equally, the cost of evaluating the kernel function does not affect the complexity as it is limited independently of the number of training systems. This is a consequence of the upper bound on the dimension of the atomic environments (cf. Subsection 5.2.1). Additionally, the calculation of the atomic environments themselves has a complexity of $\mathcal{O}(K) = \mathcal{O}(N)$ as described in Section 5.2.2.

Multiplication with the sparse matrix L in step 2 as well as the matrix-vector multiplication in step 5 has a computational complexity of $\mathcal{O}(N^2)$, as the number of entries in each row of L is independent of N, whereas steps 3 and 4 cost $\mathcal{O}(N)$. The most costly step is hence the inversion of the perturbed covariance matrix, step 4, which has the complexity of $\mathcal{O}(N^3)$ for an $N \times N$-matrix.

This inversion determines the overall complexity of the learning of the training data which remains at $\mathcal{O}(N^3)$. While this cost is interesting to note, it plays only a minor role in practice as the generation is done only once. The more important information is how much it costs to evaluate the potential for a given particle system, which we analyse next.

Prediction of Energy Values and Forces

In order to predict the energy for a new particle system in our atomic decomposition ansatz, we have to evaluate Equation (3.10) once for each particle, hence $P^\star$ times. The coefficients have been calculated before-hand and are independent of the particular test system. The evaluation cost of Equation (3.10) itself is independent of the number of particles $P^\star$. It only depends on the number of atomic systems present in the training data, which is considered as constant at this point. Of course, we need to consider the computational cost of calculating the atomic environments as well. As noted before, this scales equally as $\mathcal{O}(P^\star)$.

The argumentation remains valid for the prediction of forces, given by equation (3.20).

We conclude that the complexity of the prediction as a function of the number of particles is given by $\mathcal{O}(P^\star)$, meaning our framework scales linearly in the number of particles. This is very important in order to be able to tackle large scale problems with thousands of particles.

5.4. Evaluation Procedure

The aim of this thesis is not only to present a new method for the generation of atomic potentials, but to also assess its performance. To this end, we subject it to a validation procedure.

First of all, we have to decide how to measure the quality of a prediction $f(x)$ for a given instance x and target value y, by defining a suitable loss function $c : \mathcal{X} \times R \times R :\to [0, \infty)$. In our analysis we will use both the ℓ_1 error, i.e. the absolute difference, and the ℓ_2 error, the squared difference, as loss functions,

$$\begin{aligned} c_{\ell_1}(x, y, f(x)) &= |y - f(x)|, \\ c_{\ell_2}(x, y, f(x)) &= (y - f(x))^2. \end{aligned} \tag{5.4}$$

We now obtain the empirical error for a given test set $X^\star = (x_m^\star)_{m=1}^M$ by summing over the loss function values of each instance and normalising by their number,

$$R_{empirical}(X^\star) = \frac{1}{M} \sum_{m=0}^{M} c(x_m^\star, y_m^\star, f(x_m^\star)). \tag{5.5}$$

Plugging in c_{ℓ_1} and c_{ℓ_2} results in the mean absolute error (MAE) and root mean squared error (RMSE), respectively,

$$\begin{aligned} MAE(X^\star) &= \frac{1}{M} \sum_{m=0}^{M} |y_m^\star - f(x_m^\star)|, \\ RMSE(X^\star) &= \frac{1}{M} \sum_{m=0}^{M} (y_m^\star - f(x_m^\star))^2. \end{aligned} \tag{5.6}$$

These two errors are well established in the statistics community for assessing the prediction accuracy of regression models [57]. In our case we calculate them for the prediction of the energy values. As we have only total energies as target values, we are forced to sum up the predicted atomic contributions when calculating the difference.

When evaluating on the silicon data sets, the prediction errors are calculated per atom, as it is often done in the material science community for better comparability independent of supercell sizes. To achieve this, we divide both the target and the predicted energy by the number of atoms present in the supercell.

Of course, the real goal is not the calculation of the errors on a set where target values are known, but an assessment of the transferability of the prediction to new data points. In order to obtain a estimate of the generalisation error on unseen test examples, we employ k-fold cross validation on our supervised data set, which will be described in the next section.

A pseudo-code summary of the evaluation procedure as implemented in this thesis to obtain the numerical results presented in Chapter 6 can be found in

Listing 5.2. The presented method `evaluate()` takes as input the prepared data set, which has the local atomic enviroments already calculated, the number of cross validation runs, and the parameters of the Gaussian process regression: the variance of the noise, the kernel and its hyperparameters or suitable candidates for model selection.

5.4.1. k-fold Cross Validation

k-fold cross validation is a standard procedure to estimate the generalisation error when only limited data sets are available. The case of $k = 2$ was formally introduced by Stone in 1974 [52]. Since then, it has been widely used by the Machine Learning community as a simple yet effective method for both performance evaluation and model selection. We will address the latter aspect of model selection in Subsection 5.4.2. For an analysis of the statistical properties of k-fold cross validation, we refer to the publications [26] and [44] and the references therein.

In detail, the procedure works as follows. The data set is randomly partitioned into k equally sized *folds* and in each of the k runs one fold is withheld for testing while the potential is learned on the remaining $k - 1$ folds. This way k models are built and the MAE and RMSE (in our case of the total energy) are calculated over all instances in the particular test set for each model. The mean value and the standard deviation of the distributions of both errors are then determined over all cross validation runs. These are the final prediction errors stated in the results chapter.

As the test set cycles systematically through all folds, every instance in the data set is used once for testing and $k - 1$ times for learning. The higher the number of splits k is chosen, the more stable the training sets are and the smaller the variance of the estimate becomes. It can be shown that for the maximum number of splits possible, i.e. with only one instance in each test set, the prediction becomes an almost stable estimator of the generalisation error [47]. As a trade-off between stability of the estimate and computational cost, we choose values of k between 5 and 10. We are aware that this means that our findings constitute a proof of concept and could be enforced more rigorously by investing more calculation time.

The variance of the estimate could also be reduced when stratifying the cross validation by making sure that the distribution of target values in each training set resembles the distribution over the complete data set. We will refrain from this extra preparation step but keep it in mind as another possible future refinement of the validation procedure.

5.4.2. Selection of the Hyperparameters

Until now we have only described the evaluation routine for the LC-GAP without taking into account that the predictive power depends strongly on

```
evaluate(data_set,
         n_cvruns,
         noise,
         kernel,
         hyperparameters,
         candidates):
    # load data
    AS = load_data(data_set)
    # make splits for CV
    n_fold = floor(len(AS)/n_cvruns)
    ind = arange(0, len(AS))
    P = random.permutation(ind).reshape(n_cvruns, n_group)
    # do the CV runs
    for test_indices in P:
        # split data
        test_systems = AS[test_indices]
        training_systems = AS[!test_indices]
        # do model selection via nested CV if necessary
        if ms_nested_cv:
            hyperparameters = nested_cv(training_systems,
                                        n_cvruns-1,
                                        noise,
                                        kernel,
                                        candidates)
        # specify kernel (example)
        k = mgauss.IsotropicSquaredExponential(hyperparameters)
        # do model selection via optimisation if necessary
        if ms_likelihood:
            opt = mgauss.HyperparameterOptimizer()
        else:
            opt = None
        # learn training data
        td = mgauss.TrainingData()
        td.setVariance(noise)
        for system in traing_systems:
            td.addLocalizedData(system.local_environments,
                                system.energy)
        # generate Gaussian Approximation
        gd = mgauss.GaussianApproximation.generate(td, k, opt)
        # test
        for system in test_system:
            for environment in system.local_environments:
                predicted_energy += gd.evaluate(environment)
            absolut_diff[system.index] = abs(system.energy-
                predicted_energy)
        # calculate errors of the particular CV run
        MAE[cv_run] = mean(absolut_diff)
        RMSE[cv_run] = sqrt(mean(absolut_diff**2))
     # calculate average over all CV runs
     overall_mean_MAE = mean(MAE)
     overall_std_MAE = std(MAE)
     overall_mean_RMSE = mean(RMSE)
     overall_std_RMSE = std(RMSE)
```

Listing 5.2: Pseudo-code description of the evaluation procedure using k-fold cross validation.

well-chosen hyperparameters of the underlying Gaussian process. We will now present the two methods we employ for model selection, i.e. the procedure of choosing the parameter setting that is expected to have the best transferability to unseen input configurations based on the available training data. The first is nested cross validation, the other maximisation of the marginal likelihood.

In our framework, we only recognise parameters of the kernel, e.g. the amplitude and the characteristic length scale for the isotropic Gaussian, as hyperparameters that need to be fitted to the training data. Contrary to many others including Bartók *et al.* [4], we do not include the variance of the Gaussian noise. Instead we interpret it as fixed by the numerical instability of the method used to generate the data. For the silicon data set where we calculated the DFT energies ourselves we set the noise to the default tolerance of the numerical solver of the DFT calculator, $2.7 * 10^{-3}$ eV. For the biomolecular data sets, however, we do not have any information about the uncertainty of the provided atomisation energies. Here, we set it to the default value of 10^{-6} kcal/mol.

Using Nested Cross Validation

Nested cross validation as a model selection procedure has been extensively studied, see e.g. [11] and the references therein.

It is possible to extend the evaluation via k-fold cross validation to include model selection in each of the k runs. To this end, one executes a nested cross validation with $k-1$ folds on the current training set. Each candidate parameter or parameter combination is tested $k-1$ times, once on each fold, while having been trained on the remaining $k-2$ folds. Then, the candidate that performed best by leading to the smallest MAE on the average of folds, is chosen and used for training on the complete training set as part of the superior run. This means that depending on the particular cross validation run, possibly different models are selected, whose performance is then averaged into the final prediction error. It is important to not use the test set for evaluation of the performance in the model selection process as to ensure its unbiasedness.

In contrast to model selection via maximisation of the likelihood, nested cross validation is only able to select its parameters from a finite candidate list. However, this also means that larger parameter regions can be sampled and the accuracy improved by refining the candidate grid.

Using Maximisation of the Likelihood

An alternative approach to selecting the hyperparameters is to make use of the negative logarithm of the likelihood given in equation (2.26) as described in Section 2.1.1.

The optimisation routines used are local routines, meaning they depend strongly on the choice of initial values. Therefore, it is a reasonable approach to first

execute a nested cross validation based on a coarse candidate list and then refine the chosen value via local maximisation of the marginal likelihood.

6. Numerical Results

In this chapter, the numerical results obtained for the *Localised Coulomb matrix based Gaussian Approximation Potential (LC-GAP)* are presented. The description is split into two parts based on the data sets studied. First, we consider the biomolecular data sets in Section 6.1 and then the crystal data set for silicon in Section 6.2. Prediction errors are stated as mean absolute error (MAE) ± standard deviation to the reference energy value calculated using DFT methods, as well as root mean squared error (RMSE) ± standard deviation. As units we use kcal/mol for the biomolecules and meV per atom for the crystals. In the summarising tables throughout this chapter, we highlight important results using a boldface font. For the details concerning preparation of the data sets, model selection and cross validation (CV) procedures, refer to the preceding Chapter 5.

Starting with the assessment of the localised Coulomb matrices on subsets of the data set QM7, on which the global variant was introduced by Rupp *et al.* in [45], we evaluate slightly different variants as to their performance and cost. As common in Machine Learning applications, we use the isotropic Gaussian kernel as the standard covariance measure, but also investigate the benefit of the anisotropic variant, which is able to weigh diagonal and off-diagonal parts of the localised Coulomb matrices independently. Furthermore, we assess the validity of the localisation ansatz by intensifying penalisation of the distance to the central atom and by varying the cutoff parameter which limits the size of the atomic neighbourhood. We test a potential trained on a subset of small molecules on a superset consisting of larger molecules in order to testify the transferability of the method. Additionally, we present a saturation study on the original data set QM7.

In Section 6.2, we describe the results of applying the LC-GAP to silicon crystal structures. We analyse the performance when learning minima in the PES and their neighbourhood by sampling across all degrees of freedom. As an important application for molecular dynamics simulation, we test the prediction of gradient values in addition to the total energy.

6.1. Results on Biomolecular Data

In [20], Hansen *et al.* validate different Machine Learning techniques for predicting molecular atomisation energies. The methods they study include kernel ridge regression with isotropic Gaussian kernels. It is equivalent to Gaussian

Process regression combined with an isotropic Gaussian covariance function (cf. Section 2.3). Using the whole QM7 data set and Coulomb matrices as a global descriptor, they achieve a mean absolute error of 8.57 ± 0.40 kcal/mol for a regularisation parameter of $\lambda = 1.67 * 10^{-7} \pm 0.00$. As for the characteristic length scale of the Gaussian kernel, they obtain a mean value of 77 ± 0 via 4-fold nested cross validation using their global ansatz.

Keeping in mind that the regularisation parameter is equivalent to the variance of the Gaussian noise in the GP framework, this is the result we want to compare with. To this end, we evaluate the LC-GAP on the subsets QM4, QM5 and QM6. For better comparability, we state the resulting mean absolute errors as done by Hansen *et al.*. In the summarising tables, however, the RMSEs are provided as well for completeness. Unless stated otherwise, all results were obtained for an assumed noise variance of $\sigma_\varepsilon^2 = 10^{-6}$ kcal/mol.

6.1.1. Comparing Different Variants of the Localised Coulomb Matrix

As described in Chapter 4, there are multiple minor choices to be made when defining a localised Coulomb matrix. First of all, we test the standard variant as defined in Equation (4.9). The obtained MAEs are 6.73 ± 1.73 , 6.03 ± 1.31 and 4.23 ± 0.22 kcal/mol respectively, as can be seen in Table 6.1. This means that our results are of comparable accuracy to those of Hansen *et al.* even on the smallest data set QM4 which has only 44 molecules in each training set when executing 5-fold cross validation. On the also small data set QM5 (172 molecules in each training set), we obtain a slightly improved MAE, which is further reduced on the QM6 data set featuring 932 molecules. It is noteworthy that our results were obtained using regular cross validation (all training set combinations are totally random). In contrast, Hansen *et al.* stratified the data set, making sure that in each training set combination molecules spanning the complete range of atomisation energies were present. It stands to reason that our results would further improve, should we include this extra preprocessing step.

The values of the characteristic length scale of the isotropic Gaussian kernel, that were selected via a nested cross validation routine, are 23.40 ± 1.96 (for QM4), 22.60 ± 3.20 (for QM5) and 14.60 ± 1.96 (for QM6, averaged over all cross validation runs). They are considerably smaller when compared to the value of Hansen *et al.* cited above, resulting in more localised Gaussian kernel functions, in accordance with the framework.

When extending the model selection procedure to include maximisation of the marginal likelihood on the data sets QM4 and QM5, the results neither improve nor deteriorate significantly, cf. Table 6.2. The characteristic length scales found do not deviate much from the initial values chosen by the cross validation, meaning they are already good approximations to the (local) minima. Even including the amplitude in the model selection procedure does not improve the result in a

significant manner. This is to be expected, since the amplitude is a linear factor of the kernel. Based on the definition of GP regression, the final functional form is a linear combination of kernel functions centered at the training data points, hence all changes to the amplitude only affect the coefficients calculated by the regression but not the overall quality of the representation. The unimproved accuracy of the prediction does not justify the large additional computational cost required for the local optimisation routines. Extending the optimisation to using global methods would be simply unfeasible. Therefore, we deem the hyperparameters chosen via nested cross validation accurate enough and refrain from executing the minimisation of the negative marginal likelihood on every run.

Although the above results are already quite satisfactory, we compare them to the performance of slightly different variants, since there is no unique way to define a localised Coulomb matrix. We now test the alternative definition with the non-weighted diagonal as defined in Equation (4.10), abbreviated by adding WLD (weight-less diagonal). This results in larger errors and larger characteristic length scales compared with the standard variant, as can be seen in Table 6.3. Evaluation on QM5 gives a MAE of 8.99 $\pm$ 1.11 kcal/mol for a mean characteristic length scale of 41.00 $\pm$ 9.80, whereas using QM6 results in a MAE of 6.10 $\pm$ 1.44 kcal/mol for a mean characteristic length scale of 33.00 $\pm$ 2.53. It seems as if the decay on the diagonal, which is deliberately excluded for this variant, helps the covariance function to better distinguish between different local environments, as it introduces a continuity in the entries. We will therefore abide by the standard variant for the rest of the following analysis.

The third variant we examine is not related to the definition of the entries but the dimensions of the representation fed to the kernel. Since the localised Coulomb matrix is symmetric by construction, storing only the upper triangular matrix in order to reduce storage costs should not have a negative impact on the predictive power of the Gaussian process regression. Testing of the hypothesis that only redundant information is discarded, is done using the standard variant of the localised Coulomb matrix as defined in (4.9). The obtained MAEs are 6.66 $\pm$ 3.47, 5.95 $\pm$ 1.58 and 4.68 $\pm$ 0.51 kcal/mol respectively, (cf. Table 6.4). The model selection via cross validation leads to slightly smaller characteristic length scales of 17.50 $\pm$ 1.32 (QM4), 15.00 $\pm$ 2.00 (QM5) and 9.80 $\pm$ 1.60 (QM6), while still achieving an accuracy comparable to the procedure using the whole matrices.

From now on, we will profit from the reduced cost of the learning process stemming from the reduced input dimensions and use this trimmed variant for the representation of the local atomic environments in our further studies. We will identify it with the abbreviation Local Coulomb matrix UT (upper triangular).

In order to conclude the comparison of the performance of the different localised Coulomb matrix variants, we determine the respective MAEs for training set sizes ranging from 100 to 900 molecules using QM6. As before, we use 5-fold

Data Set	Training Set Size	Characteristic Length Scale	MAE [kcal/mol]	RMSE [kcal/mol]
QM4	44	23.40 ± 1.96	6.73 ± 1.73	8.77 ± 2.47
QM5	172	22.60 ± 3.20	6.03 ± 1.31	9.22 ± 2.39
QM6	932	14.60 ± 1.96	**4.23 ± 0.22**	7.47 ± 1.19

Table 6.1.: Prediction errors using the localised Coulomb matrix (variance Gaussian noise: 10^{-6} kcal/mol; No. CV runs: 5; grid char. length scale: 1:4:100, 1:4:50 (QM6); amplitude Gaussian kernel: 1.0)

Data Set	Train. Set Size	Amplitude	Char. Length Scale	MAE [kcal/mol]	RMSE [kcal/mol]
QM4	44	49.24 ± 1.56	21.02 ± 0.63	7.03 ± 1.22	9.13 ± 1.41
QM5	172	51.02 ± 1.47	19.29 ± 0.32	6.06 ± 0.88	10.18 ± 1.31

Table 6.2.: Prediction errors using the localised Coulomb matrix (variance Gaussian noise: 10^{-6} kcal/mol; No. CV runs: 5)

Data Set	Training Set Size	Characteristic Length Scale	MAE [kcal/mol]	RMSE [kcal/mol]
QM4	44	52.20 ± 20.61	11.36 ± 2.41	17.45 ± 3.93
QM5	172	41.00 ± 9.80	8.99 ± 1.11	15.46 ± 5.06
QM6	932	33.00 ± 2.53	**6.10 ± 1.44**	16.99 ± 12.35

Table 6.3.: Prediction errors using the localised Coulomb matrix WLD (variance Gaussian noise: 10^{-6} kcal/mol; No. CV runs: 5; grid char. length scale: 1:4:100; amplitude Gaussian kernel: 1.0)

Data Set	Training Set Size	Characteristic Length Scale	MAE [kcal/mol]	RMSE [kcal/mol]
QM4	49	17.50 ± 1.32	6.66 ± 3.47	8.13 ± 3.89
QM5	189	15.00 ± 2.00	5.95 ± 1.58	9.62 ± 3.29
QM6	932	9.80 ± 1.60	**4.68 ± 0.51**	9.73 ± 2.19

Table 6.4.: Prediction errors using the localised Coulomb matrix UT (variance Gaussian noise: 10^{-6} kcal/mol; No. CV runs: 8 (QM4, QM5), 5 (QM6); grid char. length scale: 1:4:100; amplitude Gaussian kernel: 1.0)

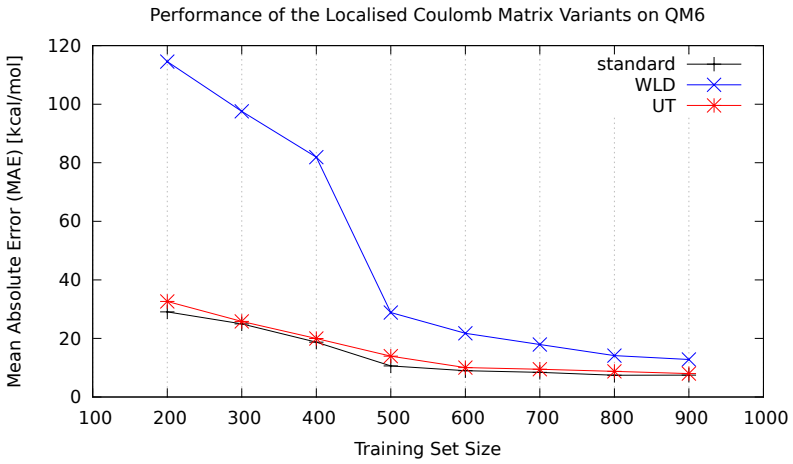


Figure 6.1.: Plot of the mean absolute errors obtained for the localised Coulomb matrix variants: standard, WLD and UT.

cross validation. As hyperparameters we choose the values selected via nested cross validation as indicated in the Tables 6.1, 6.3 and 6.4. The results are plotted in Figure 6.1 as a function of the training set size for all three variants.

We clearly observe the following two trends. Firstly, the weight-less diagonal variant does significantly worse than the standard variant. It will therefore not be considered for the rest of the analysis. Secondly, storing only the upper triangular part of the localised Coulomb matrix does not interfere with its predictive power.

6.1.2. Using a Higher Degree of Localisation

Enforcing the localisation of the respresentation of the atomic environments can be done by different means. One possibility is to penalise the distance to the central particle more severely, using a higher exponent α in the denominator of the entries of the localised Coulomb matrices as described in Equation (4.16). We test the effect of this approach by setting α to 6.0, inspired by the Lennard-Jones exponent. Evaluation is done on both the normal data sets QM4, QM5 and QM6, as well as on the reduced data sets QM5_onlyC, QM6_onlyC and QM7_onlyC. From those, all molecules including atoms other than carbon and hydrogen have been excluded in order to rule out any dipoles possibly leading to long-range interactions. Refer to Section 5.1 for a detailed description of the preparation of the data sets.

There are two surprising trends apparent when comparing the results presented in Table 6.5 with those obtained using the standard exponent of the localised Coulomb matrix. First of all, the MAEs are significantly smaller but slightly more spread (relatively seen), as we have a MAE of 3.95 ± 2.57 kcal/mol for QM4 and a MAE of 2.32 ± 0.81 kcal/mol for QM5. It is interesting to note that the MAE itself improves only slightly on the larger data sets, as it is already

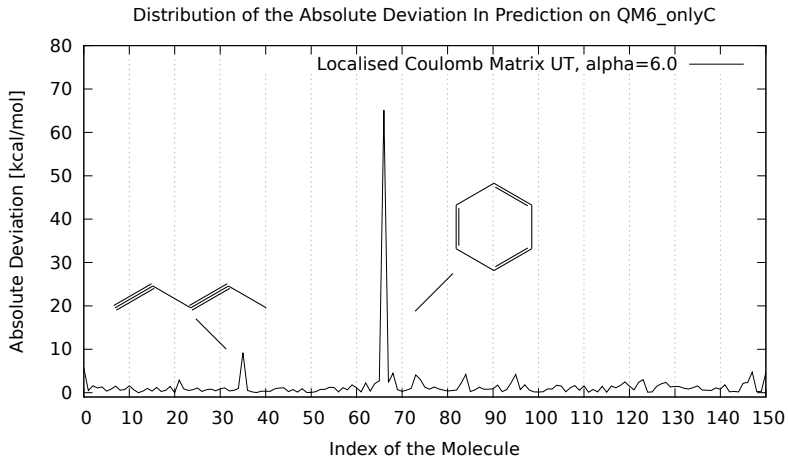


Figure 6.2.: Absolute deviation in total energy for all molecules in QM6 using the localised Coulomb matrix UT, $\alpha = 6.0$. The outlier on the left corresponds to 1,3-pentadiene and the one on the right to benzene. Both are depicted by their skeletal formula.

rather small for QM4, but the deviation diminishes. Still, the improvement from QM4 to QM6 is about 3 kcal/mol as we obtain a MAE of 1.62 ± 0.15 kcal/mol for QM6. Secondly, the chosen characteristic length scales are significantly smaller, ranging from 0.70 ± 0.35 for QM4 to 0.20 ± 0.03 for QM6.

These trends carry over to the filtered data sets. Here, the MAEs are even more spread (cf. Table 6.6), leading to MAEs of 6.86 ± 10.86 kcal/mol for QM5_onlyC, 2.52 ± 1.48 kcal/mol for QM6_onlyC and 2.73 ± 1.75 kcal/mol for QM7_onlyC.

Inspecting the data set for the reason of this large standard deviation, one can identify a handful of molecules, for which the prediction is significantly worse than for the rest. In order to illustrate this, the distribution of the absolute deviation of the predicted energy to the target value is plotted examplarily for the data set QM6_onlyC in Figure 6.2. One can easily identify two outliers, for which the prediction is significantly worse than in average; one very strong at index 66 corresponding to benzene, and a smaller one at index 35 corresponding to 1,3-pentadiene. The situation is similar for the other data sets QM5_onlyC and QM7_onlyC. In each of them several molecules are found for which inference fails, as they feature unique configurations in the training set (mostly cycles or triple bonds), leading to an undersampling effect. For a complete list of these outliers refer to Subsection 5.1.1.

Exclusion of the outlier molecules helps diminish the spread of the MAEs and produces values of 1.57 ± 0.56, 1.20 ± 0.38 and 1.56 ± 0.21 kcal/mol for QM5_onlyC*, QM6_onlyC* and QM7_onlyC* respectively.

It is a major result that the accuracy of the LC-GAP for general small biomolecules can be enhanced to a MAE of 1.62 ± 0.15 kcal/mol by raising the localisation parameter of the localised Coulomb matrix. For alcanes, it is re-

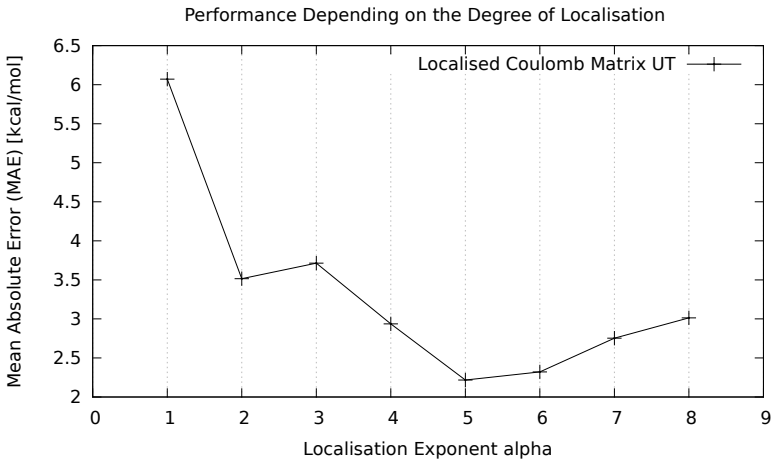


Figure 6.3.: Plot of the mean absolute error obtained for the localised Coulomb matrix UT combined with different values of the localisation exponent α on QM5

duced even further to a MAE of 1.20 ± 0.38 kcal/mol; both results nearing the desired chemical accuracy of 1 kcal/mol. In Section 6.2.1 we test if the same holds true for silicon data.

Trying to get a feel how much localisation is optimal, we test the performance of additional values for the exponent on the data set QM5. The results of preliminary testing (a systematic approach using a model selection procedure that includes different descriptor variants into a nested cross validation goes beyond the scope of this thesis and is a subject for further research) can be found in Figure 6.3 and indicate that an optimal value lies between 5.0 (MAE of 2.22 ± 0.64 kcal/mol) and 6.0 (MAE of 2.32 ± 0.81 kcal/mol). Note that raising the exponent α in Equation 4.16 to a value of 2.0 descreases the MAE already to half its value for the standard exponent of 1.0. Since the variation in MAEs is quite small, we will continue to use the value of 6.0 for testing localised Coulomb matrices with a higher degree of localisation.

6.1.3. Using an Anisotropic Gaussian Kernel

The results of the previous subsections were obtained using the isotropic Gaussian kernel, a standard choice in kernel-based machine learning methods. Isotropic means that all dimensions of the input to the kernel are weighted equally, resulting in only one characteristic length scale hyperparameter. Dropping the assumption of equal importance of all dimensions, one arrives at the anisotropic Gaussian kernel as defined in Equation (2.20), which assigns one characteristic length scale to each input dimension.

In our case the input to the kernel are the local environment descriptions, i.e. the localised Coulomb matrices. The entries of these are not defined homogeneously, hence the isotropy assumption is questionable. In order to restrict the number of hyperparameters, which have to be optimised, we distinguish

three different kinds of entries in the localised Coulomb matrices. Firstly of all, the diagonal entry belonging to the central particle which is a polynomial of its nuclear charge, secondly, the other diagonal entries which consider the interaction between the central particle and a specific neighbour, and lastly the off-diagonal entries scaling pairwise interactions with the distance to the central particle. Weighting these three different "regions" independently each with its own characteristic length scale, l_c, l_d, l_o, one hopes to enable the covariance function to improve its notion of similarity for atomic environments. The vector of characteristic length scales used to achieve this, has the form

$$\mathbf{l} = (\underbrace{l_c, l_o, ..., l_o}_{\#neighbours+1}, \underbrace{l_d, l_o, ..., l_o}_{\#neighbours}, \underbrace{l_d, l_o, ..., l_o}_{\#neighbours-1}, ..., l_d), \tag{6.1}$$

as only the upper triangular part of the localised Coulomb matrices is stored row-wise as a vector.

We test if this increased number of hyperparameters and the associated cost for the model selection reflects a better predictive power of the Gaussian process regression. This is done on the QM4 and QM5 data sets using the localised Coulomb matrix UT. We refrain from executing the evaluation procedure on the QM6 data set due to the elevated costs of selecting a combination of three hyperparameters via cross validation.

Unfortunately, the results do not show the improvement we hoped for. The MAEs obtained when allowing for these two additional degrees of freedom are not better than when using the isotropic kernel. As shown in Table 6.7, they are 6.51 ± 1.88 kcal/mol for QM4 and 5.91 ± 0.83 kcal/mol for QM5, compared to 6.66 ± 3.47 kcal/mol and 5.95 ± 1.58 kcal/mol respectively. Additionally, the model selection for the three values for the characteristic length scale is rather inconclusive. With deviations of 50.00 ± 20.98 for the length scale belonging to the diagonal entry of the central atom on QM4 or 46.00 ± 36.66 for the other diagonal entries on QM5, one is not even able to derive a tendency for a good combination for l_c, l_d, and l_o.

The same behaviour can be found when increasing the localisation of the descriptor by setting α once again to 6.0 (cf. Table 6.8).

As the results on QM4 and QM5 are less than satisfactory for both values of the exponent α, we reject the hypothesis that identifying different regions in the localised Coulomb matrix based on the different definitions of the entries leads to an improvement in comparing local atomic environments. However, we do not dismiss the promise of refinement by using an anisotropic kernel just yet. We now invest the computational cost of maximising the likelihood with respect to the whole characteristic length scale vector. We use the length scales chosen for the isotropic kernel as initial values to the local optimisation.

The prediction errors for this approach can be found in Table 6.9 for $\alpha = 1.0$ and in Table 6.10 for $\alpha = 6.0$. With MAEs of 4.04 ± 0.95 (QM4), 5.84 ± 1.51 (QM5) and 4.09 ± 0.58 kcal/mol (QM6), the results show only a slight improvement with respect to the isotropic Gaussian for the standard degree

Data Set	Training Set Size	Characteristic Length Scale	MAE [kcal/mol]	RMSE [kcal/mol]
QM4	44	0.70 ± 0.35	3.95 ± 2.57	6.10 ± 4.36
QM5	172	0.49 ± 0.24	2.32 ± 0.81	4.15 ± 2.49
QM6	932	0.20 ± 0.03	**1.62 ± 0.15**	2.90 ± 0.74

Table 6.5.: Prediction erors using the localised Coulomb matrix UT, $\alpha = 6.0$ (variance Gaussian Noise: 10^{-6} kcal/mol; No. CV runs: 5; grid char. length scale: 0.05:0.05:1.0, 0.1:0.1,1.1 (QM6); amplitude Gaussian kernel: 1.0)

Data Set	Training Set Size	Characteristic Length Scale	MAE [kcal/mol]	RMSE [kcal/mol]
QM5_onlyC	40	0.08 ± 0.02	6.86 ± 10.86	17.95 ± 32.34
QM5_onlyC*	40	0.08 ± 0.02	1.57 ± 0.56	2.31 ± 0.87
QM6_onlyC	125	0.21 ± 0.17	2.52 ± 1.48	8.21 ± 7.35
QM6_onlyC*	125	0.13 ± 0.07	1.20 ± 0.38	2.01 ± 0.94
QM7_onlyC	415	0.14 ± 0.06	2.73 ± 1.75	11.07 ± 14.86
QM7_onlyC*	415	0.07 ± 0.02	**1.56 ± 0.21**	3.26 ± 1.49

Table 6.6.: Prediction errors using the localised Coulomb matrix UT, $\alpha = 6.0$ (variance Gaussian noise: 10^{-6} kcal/mol; No. CV runs: 5; grid char. length scale: 0.05:0.05:1.0; amplitude Gaussian kernel: 1.0)

Data Set	Train. Set Size	Length Scale Central Atom	Length Scale Diagonal	Length Scale Off Diagonal	MAE [kcal/mol]	RMSE [kcal/mol]
QM4	44	50.00 ± 20.98	12.00 ± 4.00	82.00 ± 9.80	6.51 ± 1.88	8.91 ± 3.80
QM5	172	14.00 ± 8.00	46.00 ± 36.66	22.00 ± 9.80	5.91 ± 0.83	10.74 ± 4.79

Table 6.7.: Prediction errors using the localised Coulomb Matrix UT and an anisotropic kernel, $\alpha = 1.0$ (variance Gaussian noise: 10^{-6} kcal/mol; No. CV runs: 5, grid char. length scales: $[10 : 10 : 100]^3$ (QM4), $[10 : 20 : 100]^3$ (QM5); amplitude Gaussian kernel: 1.0)

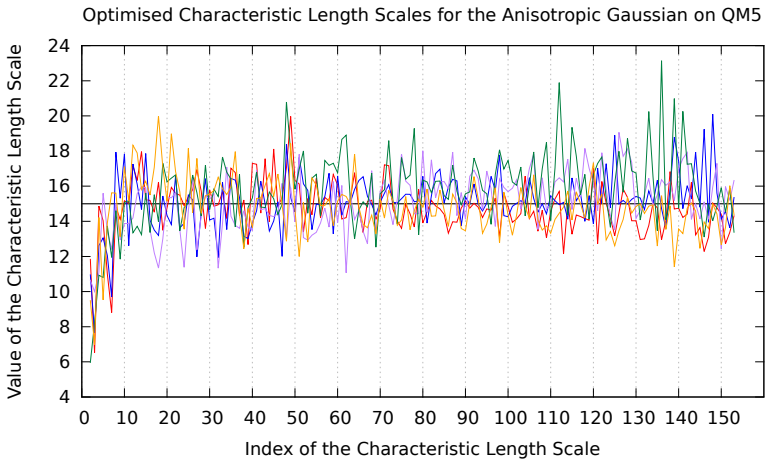


Figure 6.4.: Plot of the characteristic length scales chosen by optimisation of the marginal likelihood for the anisotropic Gaussian kernel for $\alpha = 1.0$

of localisation. For $\alpha = 6.0$ the results are even worse than when using the isotropic kernel, featuring MAEs of 3.89 ± 1.23 (QM4), 4.57 ± 2.74 (QM5) and 3.05 ± 0.55 (QM6). This could of course be due to the ever present statistical uncertainty incorporated into the evaluation procedure using cross validation with only five runs. Additionally, it is quite prossible that the local optimisation only worked rudimentarily for the high-dimensional length scale vector of 105, 153 or 210 dimensions for QM4, QM5 and QM6, as the maximum number of iterations set to 1000 was attained during the optimisation procedure.

Nevertheless, we want to analyse the hyperparameters chosen by maximisation of the likelihood for a tendency concerning distribution along the entries. To this end, we plot the values chosen on QM5 for $\alpha = 1.0$ in five cross validation runs against the indices as seen in Figure 6.4. We refrain from inferring any tendencies from the data set QM4 due to the small training set size.

Interestingly, the overall behaviour of the anisotropic characteristic length scales selected on QM5 seems to indicate that the length scales corresponding to the first 10 or so input dimensions are chosen significantly smaller than the rest of the 153 dimensions. In order to understand the reason for this effect, we plot the size of the entries of the localised Coulomb matrices for $\alpha = 1.0$ in Figure 6.5 for comparison. Due to the row-wise storage of only the upper triangular matrix combined with the sorting of the rows and columns decreasingly according to their row norms, we observe several peaks rapidly diminishing in size for larger indices. This means that the vast majority of the entries are relatively small and contribute little to the overall mass of the localised Coulomb matrices. Hence, it makes sense that the model selection via maximisation of the likelihood assigns a rather noisy but uniform characteristic length scale to over ninety percent of the entries and only distinguishes the first few entries. We conclude that the limited number of larger entries is the reason the anisotropic kernel is not able to improve the prediction significantly, as the assumption of equal importance of the input dimensions is violated only for a negligible percentage.

Data Set	Train. Set Size	Length Scale Central Atom	Length Scale Diagonal	Length Scale Off Diagonal	MAE [kcal/mol]	RMSE [kcal/mol]
QM4	44	0.10 ± 0.00	0.46 ± 0.44	0.56 ± 0.33	3.52 ± 1.37	6.44 ± 2.71
QM5	172	0.64 ± 0.44	0.50 ± 0.41	0.56 ± 0.34	2.34 ± 0.54	4.62 ± 1.72

Table 6.8.: Prediction errors using the localised Coulomb matrix UT and an anisotropic kernel, $\alpha = 6.0$ (variance Gaussian noise: 10^{-6} kcal/mol; No. CV runs: 5 , grid char. length scales: $[0.1 : 0.1 : 1.0]^3$; amplitude Gaussian kernel: 1.0)

Data Set	Train. Set Size	Initial Value Length Scale	Amplitude	MAE [kcal/mol]	RMSE [kcal/mol]
QM4	44	17.5	38.94 ±1.03	4.04 ± 0.95	5.12 ± 1.19
QM5	172	15.0	40.51 ±2.83	5.84 ± 1.51	10.46 ± 4.26
QM6	932	9.8	26.92 ±1.23	**4.09 ± 0.58**	8.37 ± 4.22

Table 6.9.: Prediction errors using the localised Coulomb matrix UT and an anisotropic kernel, $\alpha = 1.0$ (variance Gaussian noise: 10^{-6} kcal/mol; No. CV runs: 5)

Data Set	Train. Set Size	Initial Value Length Scale	Amplitude	MAE [kcal/mol]	RMSE [kcal/mol]
QM4	44	0.7	1931.78 ± 525.57	3.89 ± 1.23	6.73 ± 3.24
QM5	172	0.5	64.02 ± 15.48	4.57 ± 2.74	13.68 ± 15.97
QM6	932	0.2	14.25 ± 2.18	**3.05 ± 0.55**	11.69 ± 6.34

Table 6.10.: Prediction errors using the localised Coulomb matrix UT and an anisotropic kernel, $\alpha = 6.0$ (variance Gaussian noise: 10^{-6} kcal/mol; No. CV runs: 5)

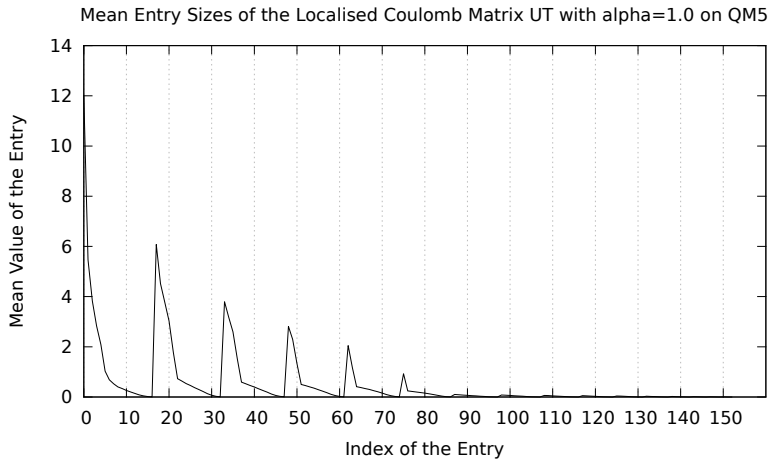


Figure 6.5.: Plot of the mean entry sizes for the localised Coulomb matrix UT with $\alpha = 1.0$ on QM5

The effect of mostly uniformly small matrix entries is strengthened by increasing the localisation exponent α to 6.0. Therefore, it is comprehensible that the anisotropic kernel does even less to improve the prediction for the more localised Columb matrices UT, as seen in Table 6.10.

We conclude that due to the strong decay in its entries, the localised Coulomb matrix is not able to profit from the higher flexibility of the anisotropic kernel in a way that would justify the additional computational cost. We will therefore continue using the isotropic variant.

6.1.4. Learning on QMx, Testing on QMy

Another reason for choosing a localisation ansatz is the hope for a wide transferability of the generated potential. Once one has learned enough different atomic environments, one would expect to be able to predict even much larger molecules. In order to verify this hypothesis, potentials are trained on the data sets QM4, QM5 and QM6 and tested on a sample of 100 other molecules from the larger data sets. As a descriptor the standard variant of the localised Coulomb matrix is used in combination with both $\alpha = 1.0$ and $\alpha = 6.0$.

The results for $\alpha = 1.0$ are not satisfactory. Whilst their error decays from QM4 to QM5 when testing on QM7d (cf. Table 6.11), it grows surprisingly for a potential built using QM6, resulting in a MAE of 499.36 ± 7.61 kcal/mol. All in all, it seems that the standard variant is not localised enough to deal with the much larger number of QM7 molecules (> 7000) and the resulting variety of environments present.

Using a higher degree of localisation, namely an exponent α of 6.0, however, causes the prediction errors to diminish greatly and continuously (cf. Table 6.12). With only the 217 QM5 molecules learned, the molecules in QM7 featur-

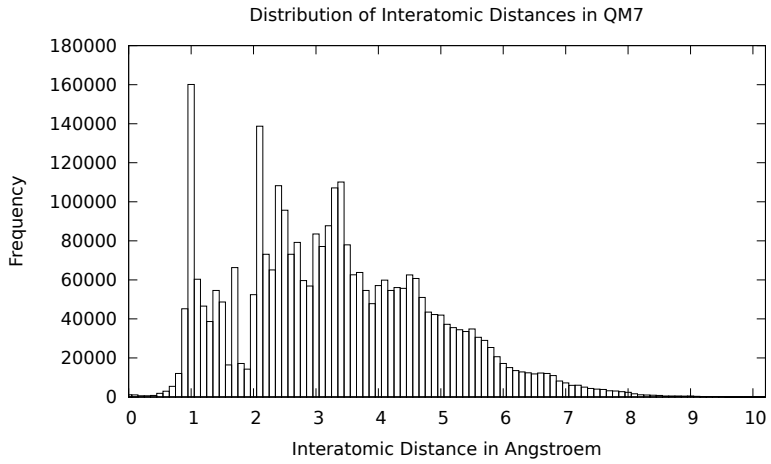


Figure 6.6.: Histogram of the interactomic distances in the data set QM7

ing up to six atoms more can be predicted with a MAE of 7.67 ± 1.12 kcal/mol. This is of the same accuracy as the results obtained by Hansen *et al.*, when training on QM7 itself [20]. Using the 1167 QM6-molecules results in a even smaller MAE of 5.22 ± 1.03 kcal/mol.

This clearly shows that potentials learned via localised GP regression are capable of transference if used in combination with a suitable local descriptor, e.g. the localised Coulomb matrix. Next, we will test the effect of introducing a finite cut-off for the atomic neighbourhoods on the performance of the LC-GAP.

6.1.5. Varying the Cutoff Parameter

Up until now, the local neighbourhood of a particle always contained all other particles composing the molecule. This was a design decision due to the manageable size of the molecules present in the data sets and the goal of comparing to the global Coulomb matrices of Rupp *et al.* introduced in [45]. For larger molecules, however, this approach is infeasible. Therefore a cut-off radius must be introduced, limiting the size of the neighbourhood.

The value of this radius has to be fixed depending on the application. For our case of biomolecules, we want to study the influence of stepwise extending the neighbourhood from encompassing only direct neighbours featuring a bond with the central particle, to including particles exhibiting a bond with a direct neighbour and so on. In order to identify suitable cut-off radii, we plot the frequencies of the interactomic distances for the largest data set QM7. The histogram can be seen in Figure 6.6.

Evidently, the first and largest peak of the distribution lies at around $1\,\mathring{A}$. This is explained by the fact that the majority of the bonds present in small biomolecules are carbon-hydrogen bonds, whose length can be measured to be 108 pm. The second peak slightly to the right of $2.0\,\mathring{A}$ represents the distance

Learned Data Set	Train. Set Size	Test Set	Char. Length Scale	MAE [kcal/mol]	RMSE [kcal/mol]
QM4	59	QM5d	23.4	37.20 ± 4.11	75.99 ± 11.53
QM4	59	QM6d	23.4	167.58 ± 9.77	189.09 ± 15.32
QM5	217	QM6d	22.6	261.80 ± 9.47	272.30 ± 9.12
QM4	59	QM7d	23.4	499.46 ± 4.48	510.49 ± 5.96
QM5	217	QM7d	22.6	333.16 ± 16.35	361.26 ± 14.68
QM6	1167	QM7d	14.6	499.36 ± 7.61	514.08 ± 6.92

Table 6.11.: Prediction errors using the localised Coulomb matrix UT, α=1.0 (variance Gaussian noise: 10^{-6} kcal/mol; No. CV runs: 5; amplitude Gaussian kernel: 1.0)

Learned Data Set	Train. Set Size	Test Set	Char. Length Scale	MAE [kcal/mol]	RMSE [kcal/mol]
QM4	59	QM5d	0.70	11.14 ± 1.08	20.62 ± 3.32
QM4	59	QM6d	0.70	19.57 ± 1.62	24.99 ± 3.94
QM5	217	QM6d	0.49	4.18 ± 0.63	8.01 ± 3.51
QM4	59	QM7d	0.70	30.01 ± 2.36	36.21 ± 3.90
QM5	217	QM7d	0.49	**7.67 ± 1.12**	14.24 ± 3.24
QM6	1167	QM7d	0.20	**5.22 ± 1.03**	16.35 ± 7.52

Table 6.12.: Prediction errors using the localised Coulomb matrix UT, α=6.0 (variance Gaussian noise: 10^{-6} kcal/mol; No. CV runs: 5; amplitude Gaussian kernel: 1.0)

between two hydrogen atoms connected by a carbon atom. Based on the frequency plot, we chose values of 2.0, 3.0 and 4.0 $\mathring{A}$ for the cut-off radius so that we include the major peaks successively. We stress that the representation of the neighbourhood using the localised Coulomb matrix is totally independent of any bond information; we only use it to derive meaningful parameters for the potential. The details of how the dimension of the localised Coulomb matrix is calculated depending on the cut-off radius can be found in Subsection 5.2.1.

We evaluate the performance of the localised Coulomb matrix combined with a cut-off radius for both the standard variant and the one with the higher localisation exponent α of 6.0. Results for the first case are found in Table 6.13 and for the latter in Table 6.14.

We first analyse the performance for the standard exponent $\alpha = 1.0$. Here, the behaviour when enlarging the cut-off radius depends on the data set. For a cutoff radius of 2.0 $\mathring{A}$ on QM4, the results are significantly better than when including the whole molecule in the neighbourhood. The MAE we obtain now is 3.86 $\pm$ 1.60 kcal/mol in comparison with 6.66 $\pm$3.47 kcal/mol. The characteristic length scale chosen via nested cross validation is larger than when not using a finite cut-off. Enlarging the atomic neighbourhood to a radius of 3.0 $\mathring{A}$ and further to 4.0 $\mathring{A}$ deteriorates the prediction accuracy on QM4 to 7.29 $\pm$ 1.01 kcal/mol and 13.16 $\pm$ 3.50. It seems that the standard variant of the localised Coulomb matrix used in combination with such a large cut-off includes too much "unnecessary" information for the small molecules present in QM4.

On QM5 using a cut-off of 2.0 $\mathring{A}$ results in a prediction accuracy of 5.50 $\pm$ 2.09 kcal/mol, which is comparable to when spanning the whole molecule. It improves slightly to 4.90 $\pm$ 0.81 kcal/mol for a radius of 3.0$\mathring{A}$ and stagnates at about this accuracy for 4.0 $\mathring{A}$ (5.01 $\pm$ 0.35 kcal/mol). Additionally, enlarging the atomic neighbourhood stabilises the characteristic length scale chosen via nested cross validation. While it is already rather well-defined for 3.0 $\mathring{A}$ (26.60 $\pm$ 4.80), it becomes clear-cut for a cut-off radius of 4.0 $\mathring{A}$ with a value of 17.00 $\pm$ 0.00.

All in all, it is important to note that the predictive power of the LC-GAP remains strong when introducing a limitation of the size of the atomic neighbourhoods on QM5. Since it does not improve significantly, however, we forego the costly evaluation for the standard variant on the additional data set QM6.

We now turn our attention to the more localised Coulomb matrix with an exponent of $\alpha = 6.0$. For those the findings improve once again significantly compared to the standard exponent, strengthening our belief that the more pronounced decline leads to a densification of the information value contained in the localised Coulomb matrix. The error values on QM4 are again already better with a MAE of 3.06 $\pm$ 1.89 kcal/mol for a cutoff radius of 2.0 $\mathring{A}$. As before, they do not improve on this data set when enlarging the cutoff, hence for the small molecules with at most 14 atoms, it is sufficient to consider a neighbourhood of 2.0 $\mathring{A}$.

On the data set QM5, which features molecules of up to 17 atoms, one notices a decline in the MAEs from 5.31 ± 2.40 kcal/mol to 2.33 ± 0.31 kcal/mol when raising the cutoff from 2.0 to 3.0 Å. It does not continue for a cutoff radius of 4.0 Å, which results in a MAE of 2.32 ± 0.39 kcal/mol. Hence, here, consideration of a 3.0 Å-environment is sufficient.

We observe the same behaviour on the data set QM6. Here, prediction is already good for a radius of 2.0 Å with a MAE of 3.03 ± 0.33 kcal/mol and improves for a spatial cutoff of 3.0 Å to 1.67 ± 0.10 kcal/mol. Then, it stagnates when enlarging the neighbourhood to 4.0 Å, with a MAE of 1.80 ± 0.28 kcal/mol.

It is interesting to note that there are only 8 atoms on average in a neighbourhood of 3.0 Å, compared to 11 atoms for the larger cutoff radius of 4.0 Å. Most surprisingly, the localised Coulomb matrix with an exponent α of 6.0 already provides significantly better results with on average only 3 atoms in the atomic environments, as is the case for a cutoff of 2.0 Å than when including the whole molecule. This once again seems to indicate that considering a too large cut-off sphere obstructs the localisation properties of our ansatz.

We conclude that the localised Coulomb matrix with α set to 6.0 is a reliable descriptor for the local atomic environments of biomolecules. It can be expected to be employed successfully in the prediction of energy values for large organic systems where the partition via a localisation ansatz is indispensable.

6.1.6. Saturation Study on QM7

We now use the findings from the previous subsections to perform a saturation study on QM7. Featuring 7165 molecules, QM7 is too large to be handled as a whole in the evaluation process at the current state of the implementation. A cross validation with 5 runs would use 5732 molecules per training set, leading to a covariance matrix with 32 million entries. To handle such large systems would go beyond the scope of this thesis. Therefore, we limit ourselves to training sets comprised of up to 1500 molecules and do not execute a model selection procedure but rather transfer the characteristic length scale l of the isotropic Gaussian kernel (2.17) of 0.2 chosen on the smaller subset QM6. For the amplitude σ_f of the Gaussian kernel, we use the default of 1.0. We set the test set size to 100 molecules which are randomly drawn for each run.

As descriptors we use the localised Coulomb matrix stored as an upper triangular matrix with a localisation exponent α of 6.0 and the spatial cutoff radius of the atomic neighbourhoods set to 3.0 Å.

The prediction errors obtained are plotted as a function of the number of training examples in Figure 6.7. We observe that the MAE decreases from about 10 kcal/mol, using only 100 training examples, to about 4.5 kcal/mol for 600 training examples, where it seems to stagnate, as there is no significant further decrease for larger training set sizes.

The RMSE, on the other hand, is much less steady and grows back to about

Data Set	Train. Set Size	Cutoff Radius [$\AA$]	Char. Length Scale	MAE [kcal/mol]	RMSE [kcal/mol]
QM4	44	2.0	48.20 ± 10.24	3.86 ± 1.60	6.96 ± 3.71
QM5	172	2.0	85.80 ± 22.40	5.50 ± 2.09	12.03 ± 7.35
QM4	44	3.0	25.80 ± 5.88	7.29 ± 1.01	11.64 ± 4.71
QM5	172	3.0	26.60 ± 4.80	**4.90 ± 0.81**	7.94 ± 2.31
QM4	44	4.0	21.80 ± 13.95	13.16 ± 3.50	19.09 ± 8.45
QM5	172	4.0	17.00 ± 0.00	5.01 ± 0.35	7.81 ± 1.16

Table 6.13.: Prediction errors using the localised Coulomb matrix UT, α=1.0 (variance Gaussian noise: 10^{-6} kcal/mol; No. CV runs: 5; grid char. length scale 1:4:100; amplitude Gaussian kernel: 1.0)

Data Set	Train. Set Size	Cutoff Radius [$\AA$]	Char. Length Scale	MAE [kcal/mol]	RMSE [kcal/mol]
QM4	44	2.0	0.97 ± 0.06	3.06 ± 1.89	4.89 ± 3.78
QM5	172	2.0	0.41 ± 0.31	5.31 ± 2.40	17.07 ± 12.56
QM6	932	2.0	0.19 ± 0.06	3.03 ± 0.33	7.70 ± 3.27
QM4	44	3.0	0.65 ± 0.35	4.05 ± 0.86	6.08 ± 1.92
QM5	172	3.0	0.60 ± 0.17	2.33 ± 0.31	4.26 ± 1.24
QM6	932	3.0	0.20 ± 0.00	**1.67 ± 0.10**	3.10 ± 0.87
QM4	44	4.0	0.70 ± 0.38	4.80 ± 4.50	8.16 ± 9.82
QM5	172	4.0	0.60 ± 0.31	2.32 ± 0.39	4.45 ± 1.86
QM6	932	4.0	0.20 ± 0.05	1.80 ± 0.28	3.76 ± 1.71

Table 6.14.: Prediction errors using the localised Coulomb matrix UT, α=6.0 (variance Gaussian noise: 10^{-6} kcal/mol; No. CV runs: 5; grid char. length scales: 0.01:0.01:1.0; amplitude Gaussian kernel: 1.0)

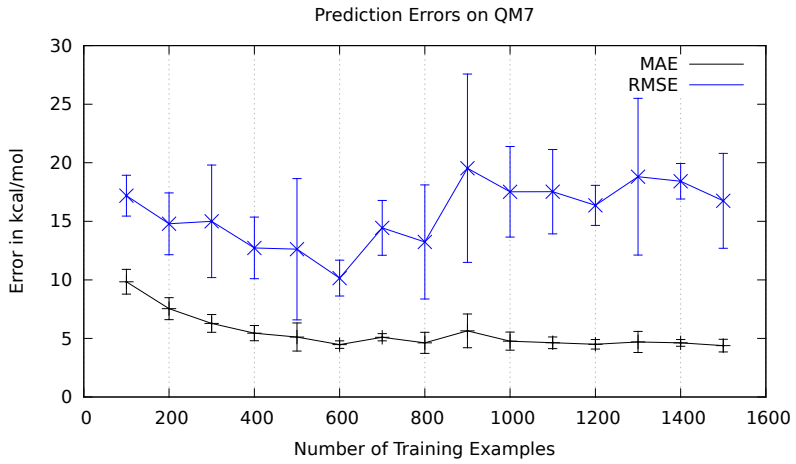


Figure 6.7.: Prediction error on QM7 using the localised Coulomb matrix UT, α=6.0, cutoff radius = 3.0 $\AA$ (variance Gaussian noise: 10^{-6} kcal/mol; No. CV runs: 5; amplitude Gaussian kernel: 1.0)

18 kcal/mol for 900 and 1300 training molecules after having decreased to about 10 kcal/mol when using 600 training examples. This is an indication of a rather large variance present in the prediction of the test sets. Due to the multitude of drastically different chemical configurations present in QM7 (such as cycloalcanes, triple-bond configurations or configurations featuring dipoles) it is to be expected that using only a limited number of training examples will not enable every configuration to be predicted with the same accuracy.

We stress once again that the mean absolute errors we obtain are better by a factor of nearly two than those reported by Hansen *et al.*, even when training on much smaller sets. They presented a mean absolute error to the DFT atomisation energy on the complete data set QM7 of 8.57 $\pm$ 0.40 kcal/mol. In contrast, our LC-GAP leads to a mean absolute error of about 4.47 $\pm$ 0.32 kcal/mol on a subset comprised of 600 randomly chosen molecules. Additionally, when constraining the data set to include only 415 alcanes with up to seven heavy atoms, we observe a mean absolute error of 1.56 $\pm$ 0.21 kcal/mol. This is already very close to the desired chemical accuracy of 1 kcal/mol, required for drug design applications.

We conclude that the localised Coulomb matrix introduced by us is a valid local descriptor at least for biomolecular applications where conformational isomers and stereoisomers are excluded.[1] In the next section, we will analyse how well it is able to cope with crystal data.

[1] **Revised Version:** The statement of the validity of the localised Coulomb matrix for biomolecular applications was restricted due to the limitations of the uniqueness of the representation for atomic environments differing only with respect to angular information.

6.2. Results on Silicon Data

In [1] and [4], Bartók-Pártay introduces Gaussian Approximation Potentials for the semiconductors carbon, silicon and germanium. Using localised GP regression with an anisotropic Gaussian kernel as the covariance function in combination with a modified bispectrum as a descriptor of atomic environments, he asserts RMSEs of less than 0.001 eV per atom in the energy. Additionally, he makes use of the ability of the Gaussian process regression to easily expand the prediction from energy to forces, for which he presents errors of less than $0.5\,\mathrm{eV}/\mathring{A}$.

As we have adopted the localised GP regression ansatz from Bartók-Pártay, we are able to assess the performance of the localised Coulomb matrix specifically in contrast to the alternative descriptor of the modified bispectrum. We restrict ourselves to silicon as the most common semiconducting material. We remark, however, that due to Bartók-Pártay adding gradient information to the training data, we can expect his prediction to be about one magnitude more accurate for a given number of training points. Incorporation of gradient information goes beyond the scope of this thesis and is an important extension that should be addressed in future research.

Since the use of an anisotropic Gaussian kernel did not improve the prediction errors significantly on the biomolecular data sets, we will refrain from investing the additional computational cost of the model selection for the strongly increased number of hyperparameters and use only the isotropic variant on the silicon data sets. We note that this decision stands in contrast to Bartók-Pártay's approach and are curious to see its impact on the results.

Bartók-Pártay selects the amplitude and the characteristic length scale of the Gaussian kernel as well as the variance of the Gaussian noise via maximisation of the likelihood. Unfortunately, he does not provide the chosen values in the supplementary information. We interpret the variance of the Gaussian noise as fixed, since it corresponds to the assumed noise in the measured data values. We use the default tolerance of $10^{-4}\,\mathrm{Hartree} = 2.7 * 10^{-3}\,\mathrm{eV}$ of the numerical solver to the DFT calculations as an approximation to the uncertainty of the DFT data. For the selection of the characteristic length scale we employ both nested cross validation and maximisation of the likelihood. When known, we try to replicate the design decisions reported by Bartók-Pártay for better comparison. Hence, the spatial cutoff is explicitly set to $4.8\,\mathring{A}$ for silicon, following Bartók-Pártay's value.

We state both MAE and RMSE in meV per atom for the energy and convert the most promising results also in kcal/mol for better comparison with the results obtained on the biomolecular data in the previous section. Note, however, that the comparison is somewhat limited for two reasons. Firstly, considering energies per atom does not make sense for systems with different atom types present, as is the case for the biomolecular data. For systems where every atom can be expected to contribute an equal amount to the total energy and

the number of atoms is constant for all test systems, it is, however, possible to convert error values per atom to error values for the total energy by multiplying with the number of atoms. This is the case in a perturbed crystal structure based always on the same supercell. Secondly, the goal of the biomolecular data sets was to show that the localised Coulomb matrix is able to handle different chemical compositions, which are all given in equilibrium configuration, leading to the possibility of learning across chemical compound space. The crystal data sets, on the other hand, put the focus on the accurate approximation of the PES of a single chemical composition including non-equilibrium states. Hence, we expect the error values to be smaller for the crystal data, at least when testing only data already close to the training systems by construction.

6.2.1. Learning Minima of the PES

When approximating the PES for a given chemical configuration the accurate prediction of local minima is most important, since they represent meta-stable states. In order to ensure this, we learn a potential using configurations located in the direct vincinity of the equilibrium configuration as training data. We propose a step-wise approach to the sampling of the high-dimensional neighbourhood exploiting that the Born-Oppenheimer PES of a crystal structure can be understood as a function of the cartesian coordinates of the atoms and the lattice vectors of the underlying supercell. In the first two steps, we perturb either only the cartesian coordinates of the atoms or only those of the lattice vectors, effectively sampling subspaces with the goal of analysing how the localised Coulomb matrix deals with them. As the last step, we perturb both atoms and lattice vectors, corresponding to a sampling across all dimensions.

Perturbing only the Atoms

We start by evaluating the localised Coulomb matrix on a data set of 8-atom silicon supercells whose atoms have been uniformly randomly perturbed by up to 0.3 $\mathring{A}$ in relation to the equilibrium configuration. This data set is abbreviated as Si8ApCC. We train on subsets with 50, 150 and 450 configurations and test on 10, 50 or 150 configurations respectively. Based on the findings on the biomolecular data sets concerning storage and permutation variants, we immediately use the upper triangular localised Coulomb matrix.

The RMS errors per atom that we obtain for the standard localisation exponent α of 1.0 are 83.40 ± 27.95, 54.98 ± 10.94 and 34.07 ± 2.48 meV for the different training set sizes as seen in Table 6.15. This is two orders of magnitude worse than the results reported by Bartók-Pártay. When naively compared to the biomolecular results, the MAE of 25.26 ± 0.39 meV per atom corresponds to a value of 4.66 ± 0.07 kcal/mol which is matchable to the performance of the localised Coulomb matrices UT on QM6. Hence our expectation of overall smaller MAEs for the silicon data is met, as we obtain comparable accuracy on a training set half the size.

Interestingly, the silicon data set requires far larger characteristic length scales for the Gaussian kernels, as it was the case for the biomolecular data sets. Model selection on a training set of 450 configurations leads to a length scale of 390.00 ± 20.00 in comparison to a value of 9.80 ± 1.60 on QM6. The significant difference in the hyperparameters reflects the ability of the LC-GAP to be fitted to any chemical configuration independent of the physical properties. This means, however, that while the LC-GAP is applicable to molecules and crystalline solids alike, potentials have to be trained for a specific chemical class in order to avoid extrapolation that is bound to fail.

We now check if selecting the hyperparameters via minimisation of the negative logarithmic likelihood improves the prediction. The values chosen via cross validation are once again used as initial values for the search for the characteristic length scale. For the amplitude the standard value of 1.0 is chosen as initial value. The results can be found in Table 6.16. While optimisation of the likelihood did not increase the accuracy significantly for the biomolecular data sets, it does so for the Si data. The RMS error is decreased to a value of 9.62 ± 0.70 meV per atom, which is only one order of magnitude worse than Bartók-Pártay's result. As noted in the introduction of this Section, these are results as good as we expect to be able to obtain without adding gradient information.

The optimisation has difficulties identifying a unique minimum for training set sizes smaller than 450 configurations. Selection using only 150 configurations results in an amplitude of 161.17 ± 135.09 and a characteristic length scale of 1861.71 ± 1450.85, both of which feature a large spread. In contrast, the values for the training set of 450 perturbed structures are 37.37 ± 0.54 for the amplitude and 459.28 ± 12.02 for the length scale.

Since raising the localisation exponent α to 6.0 improved the results significantly for the biomolecules, we test the effect also on the silicon crystals. Once again we observe a decrease in error values as well as in selected characteristic length scales for the more localised descriptors. Nevertheless, the improvement in prediction error is small when compared to the behaviour observed on the biomolecules, whereas the change in characteristic length scale is as significant as before. As summarised in Table 6.17, we obtain slightly smaller RMSEs of 56.20 ± 13.92, 37.69 ± 4.82 and 26.90 ± 7.49 meV for characteristic length scales selected to 74.00 ± 17.44, 20.00 ± 0.00 and 15.00 ± 0.00, respectively. Note that for training set sizes of 150 and 450 configurations the model selection via nested cross validation is able to uniquely identify the values.

Again, selection of the hyperparameters via optimisation of the likelihood improves the prediction (cf. Table 6.18). Using 450 training set configurations leads to a RMSE of 6.27 ± 0.24 meV. The hyperparameters of 6670.98 ± 4654.38 for the amplitude and 185.37 ± 113.31 for the characteristic length scale, are, however, not chosen uniquely over the evaluation runs. Surprisingly, this problem does not show for the smaller training set sizes.

The decision to perturb the equilibrium configuration of the silicon 8-atom supercell by up to 0.3$\mathring{A}$ was made to reproduce a data set as close as possible to

Data Set	Train. Set Size	Test Set Size	Char. Length Scale	MAE per atom [meV]	RMSE per atom [meV]
Si8ApCC	50	10	640.00 ± 228.91	63.35 ± 19.98	83.40 ± 27.95
Si8ApCC	150	50	450.00 ± 44.72	39.59 ± 5.06	54.98 ± 10.94
Si8ApCC	450	100	390.00 ± 20.00	25.26 ± 0.39	34.07 ± 2.48

Table 6.15.: Prediction errors using the localised Coulomb matrix UT, $\alpha = 1.0$ ($p_A = 0.3\,\mathring{A}$; variance Gaussian noise: $2.7 * 10^{-3}$ eV; No. CV runs: 5; grid char. length scale: 50:50:1000; amplitude Gaussian kernel: 1.0)

Data Set	Train. Set Size	Test Set Size	Amplitude	Char. Length Scale	MAE per atom [meV]	RMSE per atom [meV]
Si8ApCC	50	10	85.01 ± 22.86	1171.34 ± 293.72	45.09 ± 7.04	60.38 ± 17.78
Si8ApCC	150	50	161.17 ± 135.09	1861.71 ± 1450.85	13.50 ± 1.79	17.18 ± 2.03
Si8ApCC	450	100	37.37 ± 0.54	459.28 ± 12.02	7.42 ± 0.19	**9.62 ± 0.70**

Table 6.16.: Prediction errors using the localised Coulomb matrix UT, $\alpha = 1.0$ ($p_A = 0.3\,\mathring{A}$; variance Gaussian noise: $2.7 * 10^{-3}$ eV; No. CV runs: 5)

Data Set	Train. Set Size	Test Set Size	Char. Length Scale	MAE per atom [meV]	RMSE per atom [meV]
Si8ApCC	50	10	74.00 ± 17.44	41.91 ± 10.34	56.20 ± 13.92
Si8ApCC	150	50	20.00 ± 0.00	26.56 ± 0.86	37.69 ± 4.82
Si8ApCC	450	100	15.00 ± 0.00	18.21 ± 1.96	26.90 ± 7.49

Table 6.17.: Prediction errors using the localised Coulomb matrix UT, $\alpha = 6.0$ ($p_A = 0.3\,\mathring{A}$; variance Gaussian noise: $2.7 * 10^{-3}$ eV; No. CV runs: 5; grid char. length scale: 10:10:150, 5:5:65 (450 training examples); amplitude Gaussian kernel: 1.0)

Bartók-Pártay's specification for better comparison. Now we want to double-check the performance of the LC-GAP, by analysing the prediction errors for different perturbation values. The smaller the perturbation, the greater the accuracy in energy prediction should become, as the difference in configuration becomes less. We test this hypothesis by setting the perturbation level to 0.01, 0.1 and 1.0$\mathring{A}$ in addition to 0.3$\mathring{A}$ as done previously. For the training and test set sizes we choose 150 and 50 configurations.

The results are summarised in Table 6.19. As we expected, there is a significant decline in prediction errors for smaller perturbation values. While for a large perturbation of 1.0 $\mathring{A}$, the accuracy is bad with a RMSE of 311.95 ± 90.75 meV for an exponent $\alpha = 1.0$, it decreases to about 1 meV for a perturbation value of 0.01 $\mathring{A}$. Interestingly, the characteristic length scale decreases as well for smaller perturbation values.

We note that, contrary to our expectation, we do not observe that the prediction errors improve for a higher degree of localisation ($\alpha = 6.0$). It can be surmised that the large perturbation of 1.0 $\mathring{A}$ leads to atomic environments with a smaller number of neighbours and consequently a smaller informative value. This effect is effectively reinforced by the stronger decay of the entries of the localised Coulomb matrix due to the higher exponent in the denominator, resulting in the bad prediction error of 4761.30 ± 1657.21 meV. This hypothesis, however, does not explain why the RMSE is also slightly worse compared to the standard exponent for a perturbation value of 0.1 $\mathring{A}$ and is only comparable but not better when perturbing only by 0.01 $\mathring{A}$. We will keep an eye on the effect in the following subsections. A systematic investigation as to why the higher localisation merits such a strong improvement on the biomolecules but not on the silicon crystals, however, is beyond the scope of this thesis.

Perturbing only the Lattice Vectors

Next, we test on the data set Si8ApLV, where only the coordinates of the lattice vectors, but not of the atoms, have been perturbed by up to 0.5 $\mathring{A}$. Once again we use the upper triangular localised Coulomb matrix with both $\alpha = 1.0$ and $\alpha = 6.0$. Also, we use the same training and test set sizes as before.

Compared to the results of the previous subsection, perturbing the lattice vectors instead of the atoms seems to make accurate prediction of the total energy values more difficult. As can be seen in Table 6.20, the RMSEs when training on 50 or 150 configurations are significantly worse with values of 728.92 ± 1029.24 and 127.55 ± 21.24 meV. Using 450 training examples, the error decays to 53.95 ± 9.61 meV, which is only about twice the value obtained on Si8ApCC (34.07 ± 2.48 meV). Of course, we have to keep in mind that the perturbation acting on the lattice vectors was chosen larger than that on the atomic coordinates.

The characteristic length scales chosen via nested cross validation on the other hand are even slightly smaller, selected as 350.00 ± 31.62 for 450 training configurations.

Data Set	Train. Set Size	Test Set Size	Amplitude	Char. Length Scale	MAE per atom [meV]	RMSE per atom [meV]
Si8ApCC	50	10	194.76 ± 112.62	55.15 ± 28.57	15.07 ± 2.70	19.09 ± 4.16
Si8ApCC	150	50	87.99 ± 7.73	21.69 ± 2.27	8.32 ± 0.44	10.95 ± 1.25
Si8ApCC	450	100	6670.98 ± 4654.38	185.37 ± 113.31	4.95 ± 0.18	**6.27** ± **0.24**

Table 6.18.: Prediction errors using the localised Coulomb matrix UT, $\alpha = 6.0$ ($p_A = 0.3\,\text{Å}$; variance Gaussian noise: $2.7 * 10^{-3}$ eV; No. CV runs: 5; amplitude Gaussian kernel: 1.0)

Data Set	α	Perturb.	Char. Length Scale	MAE per atom [meV]	RMSE per atom [meV]
Si8ApCC	1.0	1.0	430.00 ± 112.25	231.77 ± 46.19	311.95 ± 90.75
Si8ApCC	1.0	0.1	300.00 ± 0.00	1.93 ± 0.27	2.44 ± 0.33
Si8ApCC	1.0	0.01	250.00 ± 0.00	0.76 ± 0.07	0.97 ± 0.09
Si8ApCC	6.0	1.0	90.00 ± 0.00	1577.04 ± 738.60	4761.30 ± 1657.20
Si8ApCC	6.0	0.1	20.00 ± 0.00	3.56 ± 0.51	4.49 ± 0.64
Si8ApCC	6.0	0.01	15.00 ± 0.00	0.77 ± 0.09	1.00 ± 0.10

Table 6.19.: Prediction errors using the localised Coulomb matrix UT while varying the perturbation (variance Gaussian noise: $2.7 * 10^{-3}$ eV; No. CV runs: 5; grid char. length scale: 50:50:1000 ($\alpha = 1.0$), 5:5:90 ($\alpha = 6.0$); amplitude Gaussian kernel: 1.0)

Data Set	Train. Set Size	Test Set Size	Char. Length Scale	MAE per atom [meV]	RMSE per atom [meV]
Si8ApLV	50	10	850.00 ± 200.00	384.76 ± 485.06	728.92 ± 1029.24
Si8ApLV	150	50	370.00 ± 24.49	86.44 ± 10.78	127.55 ± 21.24
Si8ApLV	450	150	350.00 ± 31.62	38.72 ± 3.66	53.95 ± 9.61

Table 6.20.: Prediction errors using the localised Coulomb matrix UT, $\alpha = 1.0$ ($p_{LV} = 0.5\,\text{Å}$; variance Gaussian noise: $2.7 * 10^{-3}$ eV; No. CV runs: 5; grid char. length scale: 50:50:1000; amplitude Gaussian kernel: 1.0)

Using the maximisation of the likelihood for model selection with the characteristic length scales gained via cross validation as initial values, improves the results to a RMSE of 20.89 ± 8.31 meV per atom (cf. Table 6.21). This is still twice the error value when training on 450 configurations from Si8ApCC and choosing the hyperparameters via the local optimisation routines. Interestingly, this is comparable to the RMSE obtained on Si8ApCC when using only 150 configurations for training: hence, it seems that about three times more training examples from Si8ApLV are needed to obtain the same accuracy.

When comparing the deviation of the hyperparameters chosen by nested cross validation with those selected via optimisation routines, both procedures are able to identify more or less unique values with comparable accuracy.

As in the previous subsection, intensifying the localisation improves the results only slightly (cf. Table 6.22). Training on 450 configurations leads to a RMSE of 38.96 ± 12.12 meV per atom for a characteristic length scale of 37.00 ± 19.90. We again observe the ratio of this being about twice as bad as when executing the same procedure on Si8ApCC. This behaviour carries over when using the maximisation of the likelihood for model selection (cf. Table 6.23). The RMSE decays to 15.95 ± 5.09 meV per atom for training on 450 configurations and the characteristic length scales are chosen smaller as 14.20 ± 4.56. While for the standard localisation exponent both methods of model selection produced hyperparameters with comparable accuracy, it seems that for $\alpha = 6.0$ the model selection via nested cross validation has some difficulties, leading to values with a rather relatively large spread. Maximisation of the likelihood on the other hand is able to select hyperparameters with a smaller standard deviation. On Si8ApCC it was exactly the inverse case, with the cross validation identifying unique hyperparameters instead.

We conclude that perturbation of the lattice vectors makes prediction more difficult as it introduces a larger diversity in the atomic environments present in the training and test data than when perturbing only the atoms. Hence, one needs more training examples to compensate and achieve comparable prediction errors. We expect the effect to be even more pronounced in the next subsection, where evaluation is done on a data set built by perturbing the equilibrium configuration in all possible degrees of freedom.

Perturbing both Atoms and Lattice Vectors

We now expand the perturbation of the silicon 8-atom supercell to include the cartesian coordinates of both the atoms and the lattice vectors. As before, the cartesian coordinates of the atoms are perturbed by up to 0.3 $\mathring{A}$, whereas the perturbation of coordinates of the lattice vectors is up to 0.5 $\mathring{A}$. This data set is abbreviated Si8ApALV. Concerning localised Coulomb matrix, number of training and test configurations, we use the same settings as in the previous sections for better comparison, i.e. the localised Coulomb matrix UT with a localisation exponent of 1.0 and 6.0 and 50, 150 and 450 training configurations

Data Set	Train. Set Size	Test Set Size	Amplitude	Char. Length Scale	MAE per atom [meV]	RMSE per atom [meV]
Si8ApLV	50	10	68.17 ± 7.12	745.26 ± 140.35	117.73 ± 49.21	154.66 ± 49.23
Si8ApLV	150	50	72.01 ± 14.88	706.15 ± 140.90	32.47 ± 5.57	55.73 ± 14.94
Si8ApLV	450	100	46.33 ± 0.94	419.87 ± 13.40	10.67 ± 2.82	**20.89** ± **8.31**

Table 6.21.: Prediction errors using the localised Coulomb matrix UT, $\alpha = 1.0$ ($p_{LV} = 0.5\,\mathring{A}$; variance Gaussian noise:$2.7 * 10^{-3}$ eV; No. CV runs: 5)

Data Set	Train. Set Size	Test Set Size	Char. Length Scale	MAE per atom [meV]	RMSE per atom [meV]
Si8ApLV	50	10	35.00 ± 30.33	45.27 ± 16.61	65.27 ± 26.56
Si8ApLV	150	50	56.00 ± 22.89	42.67 ± 10.32	56.51 ± 13.57
Si8ApLV	450	150	37.00 ± 19.90	29.05 ± 9.37	38.96 ± 12.12

Table 6.22.: Prediction errors using the localised Coulomb matrix UT, $\alpha = 6.0$ ($p_{LV} = 0.5\,\mathring{A}$; variance Gaussian noise: $2.7 * 10^{-3}$ eV; No. CV runs: 5; grid char. length scale: 5:5:100; amplitude Gaussian kernel: 1.0)

Data set	Train. Set Size	Test Set Size	Amplitude	Char. Length Scale	MAE per atom [meV]	RMSE per atom [meV]
Si8ApLV	50	10	78.26 ± 12.89	21.22 ± 4.17	25.93 ± 7.87	33.86 ± 10.52
Si8ApLV	150	50	96.05 ± 24.75	12.55 ± 1.12	15.40 ± 1.54	23.92 ± 3.88
Si8ApLV	450	100	310.04 ± 187.23	14.20 ± 4.56	9.01 ± 1.37	**15.95** ± **5.09**

Table 6.23.: Prediction errors using the localised Coulomb matrix UT, $\alpha = 6.0$ ($p_{LV} = 0.5\,\mathring{A}$; variance Gaussian noise:$2.7 * 10^{-3}$ eV; No. CV runs: 5)

combined with 10, 50 and 150 test cells respectively.

We start our analysis with the standard exponent, for which the results can be found in Table 6.24. As already expected, the prediction errors are worse on this data set than on both Si8ApCC and Si8ApLV. The RMSEs for 50, 150 and 450 training configurations are 471.85 ± 334.94, 219.37 ± 135.58 and 125.06 ± 71.26 meV per atom respectively. With only 50 configurations in the training set, the model selection via cross validation has significant problems identifying a suitable characteristic length scale, resulting in a value of 2500.00 ± 894.43. For comparison, training on 150 or 450 configurations leads to much smaller values with a diminished spread of 385.00 ± 20.00 and 365.00 ± 20.00. We conclude that 50 training configurations are simply not enough for a reliable model selection due to the high diversity present in this data set by construction.

Once again we use the characteristic length scales chosen by nested cross validation as initial values for the model selection via maximisation of the likelihood. The results can be found in Table 6.25. While the characteristic length scale reported as 365.45 ± 10.90 differs only in its spread from the previous runs using 450 training configurations, the RMSE is halved to 56.95 ± 8.52 meV per atom. This is comparable to the RMSEs obtained when using maximisation of the likelihood as model selection procedure on 50 Si8ApCC configurations or on 150 Si8ApLV configurations. As already suggested in the previous subsection, the combined perturbation of all degrees of freedom at once has to be compensated by learning more training examples.

In contrast to evaluation on the previous data sets, intensifying the localisation does not improve the results on Si8ApALV (cf. Table 6.26). On the contrary, the model selection via cross validation identifies uncharacteristically large length scales with a wide spread, indicating that it was not able to identify a minimum. This leads to the unsatisfactorily large RMSEs of 2302.45 ± 4121.96, 444.79 ± 307.11 and 912.82 ± 1641.91 meV per atom. Model selection via optimisation of the likelihood is able to better identify characteristic length scales, which fall into the same order of magnitude as before, e.g. 5.40 ± 0.75 for 450 training configurations. Nevertheless, (cf. Table 6.27), it obtains very large prediction errors with values of 3503.10 ± 3147.65 meV per atom when training on 450 configurations. Surprisingly, the RMSEs are smaller when using a smaller training set size, completely contradicting the idea of a converged training data set. We conclude that using a higher degree of localisation in the local Coulomb matrix renders it unable to deal with the large variety in the atomic environments due to the sampling across all dimensions of the potential energy surface.

We summarise the ability of the localised Coulomb matrix for an exponent α of 1.0 to cope with the perturbation of the different degrees of freedom (atom coordinates, lattice vector coordinates, or both) in Figure 6.8, where the prediction errors on the three data sets have been plotted as a function of the training set size.

One can clearly see the higher training set sizes required on Si8ApLV and

Data Set	Train. Set Size	Test Set Size	Char. Length Scale	MAE per atom [meV]	RMSE per atom [meV]
Si8ApALV	50	10	2500.00 ±894.43	303.72 ± 188.52	471.85 ± 334.94
Si8ApALV	150	50	385.00 ± 20.00	102.97 ± 30.69	219.37 ± 135.58
Si8ApALV	450	150	365.00 ± 20.00	57.58 ± 10.82	125.06 ± 71.26

Table 6.24.: Prediction errors using the localised Coulomb matrix UT, $\alpha = 1.0$ ($p_A = 0.3\,\mathring{A}$, $p_{LV} = 0.5\,\mathring{A}$; variance Gaussian noise: $2.7 * 10^{-3}$ eV; No. CV runs: 5; grid char. length scale: 500:500:5000 (Si8ApALV 50), 25:25:500 (Si8ApALV 150, 450); amplitude Gaussian kernel: 1.0)

Data Set	Train. Set Size	Test Set Size	Amplitude	Char. Length Scale	MAE per atom [meV]	RMSE per atom [meV]
Si8ApALV	50	10	73.36 ± 5.03	672.04 ± 58.38	179.84 ± 42.48	249.74 ± 86.36
Si8ApALV	150	50	54.89 ± 2.46	518.08 ± 27.42	61.02 ± 15.37	110.14 ± 66.89
Si8ApALV	450	100	50.41 ± 1.36	365.45 ± 10.90	32.22 ± 3.64	**56.95 ± 8.52**

Table 6.25.: Prediction errors using the localised Coulomb matrix UT, $\alpha = 1.0$ ($p_A = 0.3\,\mathring{A}$, $p_{LV} = 0.5\,\mathring{A}$; variance Gaussian noise: $2.7 * 10^{-3}$ eV; No. CV runs: 5)

Data Set	Train. Set Size	Test Set Size	Char. Length Scale	MAE per atom [meV]	RMSE per atom [meV]
Si8ApALV	50	10	150.00 ± 178.19	789.68 ± 1304.70	2302.45 ± 4121.96
Si8ApALV	150	50	220.00 ± 142.65	145.61 ± 41.34	444.79 ± 307.11
Si8ApALV	450	100	145.00 ± 121.86	179.99 ± 219.77	912.82 ± 1641.91

Table 6.26.: Prediction errors using the localised Coulomb matrix UT, $\alpha = 6.0$ ($p_A = 0.3\,\mathring{A}$, $p_{LV} = 0.5\,\mathring{A}$; variance Gaussian noise: $2.7 * 10^{-3}$ eV; No. CV runs: 5; grid char. length scale: 25:25:500; amplitude Gaussian kernel: 1.0)

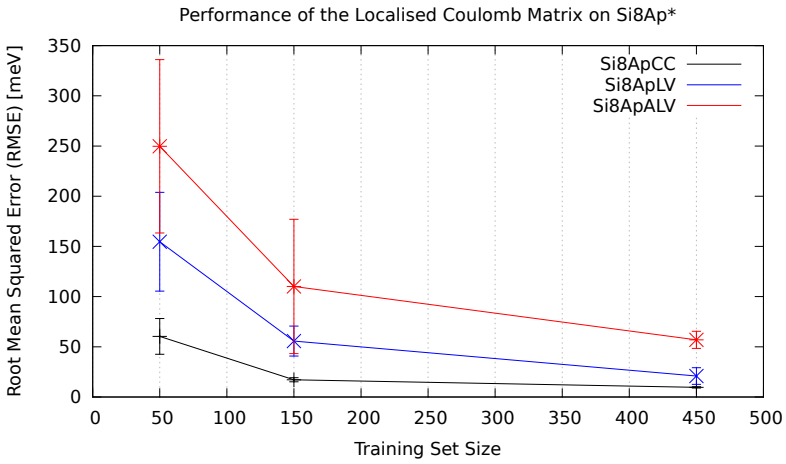


Figure 6.8.: Comparison of the root mean squared errors on the data sets Si8ApCC, Si8ApLV and Si8ApALV using the localised Coulomb matrix UT, $\alpha = 1.0$ ($p_A = 0.3\,\mathring{A}$, $p_{LV} = 0.5\,\mathring{A}$, variance Gaussian noise: $2.7 * 10^{-3}$ eV; No. CV runs: 5)

Si8ApALV to attain the same accuracy as on Si8ApCC.

6.2.2. Saturation Study on Si8ApALV

All in all, it seems that a maximum training size of 450 configurations is too small to obtain satisfactory prediction errors for a perturbation of the lattice vectors of up to $0.5\,\mathring{A}$, even with the standard localisation exponent $\alpha = 1.0$. Before investing the added computational cost for generating more training configurations and learning on larger training sets, we decrease the perturbation of the lattice values to at most $0.2\,\mathring{A}$, the same perturbation reported by Bartók-Pártay in the physical review letter introducing the GAP [4]. For the perturbation of the atomic coordinates, we stay with the value of $0.3\,\mathring{A}$.

As a higher localisation exponent did not lead to a significant improvement in accuracy on this data set, we now evaluate only using $\alpha = 1.0$.

Decreasing the maximum perturbation inflicted on the coordinates of the lattice vectors significantly improves the prediction errors, as expected (cf. Table 6.28). We now obtain a RMSE of 48.14 ± 9.85 meV per atom when training on 450 configurations. This is more than twice as good as the value obtained for a maximum perturbation of $0.5\,\mathring{A}$, which is 125.06 ± 71.26 meV per atom. It is slightly better than when training on Si8ApCC using 150 configurations, i. e. when only the atomic coordinates are perturbed. As far as the characteristic length scales are concerned, reducing the perturbation only leads to a slightly smaller value. For a perturbation of $0.5\,\mathring{A}$ 365.00 ± 20.00 was chosen as the characteristic length scale, now the value is 330.00 ± 24.49.

Using maximisation of the likelihood for model selection decreases the RMSE to 14.03 ± 0.83 meV per atom (cf. Table 6.29). This is very close to the accuracy

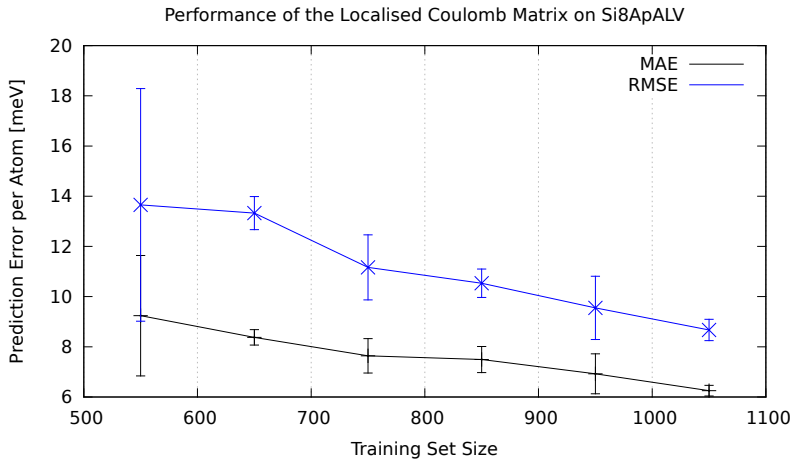


Figure 6.9.: Prediction errors on Si8ApALV as a function of the training set size using the localised Coulomb matrix UT, $\alpha = 1.0$ ($p_A = 0.3\,\text{Å}$, $p_{LV} = 0.2\,\text{Å}$; variance Gaussian noise: $2.7 * 10^{-3}$ eV; No. CV runs: 5)

of about 1 meV per atom, which is the best we expect to be possible without inclusion of gradient information. The characteristic length scale chosen via this method, 434.61 ± 8.11, is larger than when using nested cross validation. The MAE obtained on 450 training configurations is 8.76 ± 0.50 meV per atom. When converted to kcal/mol, this corresponds to a MAE for the total energy of 1.61 ± 0.09 kcal/mol. Compared to the results obtained on the biomolecular data sets under the reservations explained at the beginning of this section, this is as good as when training on either 415 alcanes or on over 900 general small organic molecules. Hence, the localised Coulomb matrix shows promising results on both finite and periodic chemical structures.

We now perform a saturation study on Si8ApALV in order to analyse the effect of the training set size on the prediction errors. To this end, we chose larger training set sizes ranging from 550 to 1050 while keeping the test set size at 100 configurations. We use the hyperparameters chosen using 450 training configurations and maximisation of the likelihood. The results are plotted in Figure 6.9.

Both MAE and RMSE show a linear decay for increased training set sizes which has not yet saturated for 1050 configurations. We explicitly remark that using more than 1000 training configurations pushes even the RMSE into the order of magnitude of 1 meV, our current benchmark goal. This clearly shows that the higher variety in local atomic enviroments on Si8ApALV can be compensated by augmenting the training set size. In [1], Bartók-Pártay also notes the large training set sizes required without specifying actual numbers. They are due to the fact that it is not possible to add only single local atomic environments because of the lack of observation of the atomic energy contributions. Hence, one has to add a complete supercell, regardless of the number of actually different neighbourhoods. In order to limit the training set sizes, Bartók-Pártay employs a sparsification process using pseudo-inputs as described by Snelson

Data Set	Train. Set Size	Test Set Size	Amplitude	Char. Length Scale	MAE per atom [meV]	RMSE per atom [meV]
Si8ApALV	50	10	112.49 ± 15.19	17.92 ± 2.35	116.85 ± 117.22	259.54 ± 376.66
Si8ApALV	150	50	181.41 ± 128.48	10.67 ± 3.46	247.79 ± 382.50	1031.91 ± 1834.94
Si8ApALV	450	100	114.03 ± 31.73	5.40 ± 0.75	486.23 ± 439.54	3503.10 ± 3147.65

Table 6.27.: Prediction errors using the localised Coulomb matrix UT, $\alpha = 6.0$ ($p_A = 0.3\,\mathring{A}$, $p_{LV} = 0.5\,\mathring{A}$; variance Gaussian noise: $2.7 * 10^{-3}$ eV; No. CV runs: 5)

Data Set	Train. Set Size	Test Set Size	Char. Length Scale	MAE per atom [meV]	RMSE per atom [meV]
Si8ApALV	50	10	920.00 ± 60.00	69.64 ± 13.17	84.08 ± 15.17
Si8ApALV	150	50	390.00 ± 66.33	54.63 ± 8.72	73.14 ± 20.52
Si8ApALV	450	150	330.00 ± 24.49	34.01 ± 2.89	48.14 ± 9.85

Table 6.28.: Prediction errors using the localised Coulomb matrix UT, $\alpha = 1.0$ ($p_A = 0.3\,\mathring{A}$, $p_{LV} = 0.2\,\mathring{A}$; variance Gaussian noise: $2.7 * 10^{-3}$ eV; No. CV runs: 5; grid char. length scale: 50:50:1000; amplitude Gaussian kernel: 1.0)

Data Set	Train. Set Size	Test Set Size	Amplitude	Char. Length Scale	MAE per atom [meV]	RMSE per atom [meV]
Si8ApALV	50	10	64.23 ± 9.39	803.23 ± 121.57	64.42 ± 9.88	82.99 ± 17.16
Si8ApALV	150	50	59.67 ± 2.90	616.30 ± 21.34	20.86 ± 1.13	30.42 ± 3.17
Si8ApALV	450	100	47.42 ± 0.88	434.61 ± 8.11	8.76 ± 0.50	**14.03 ± 0.83**

Table 6.29.: Prediction errors using the localised Coulomb matrix UT, $\alpha = 1.0$ ($p_A = 0.3\,\mathring{A}$, $p_{LV} = 0.2\,\mathring{A}$; variance Gaussian noise: $2.7 * 10^{-3}$ eV; No. CV runs: 5)

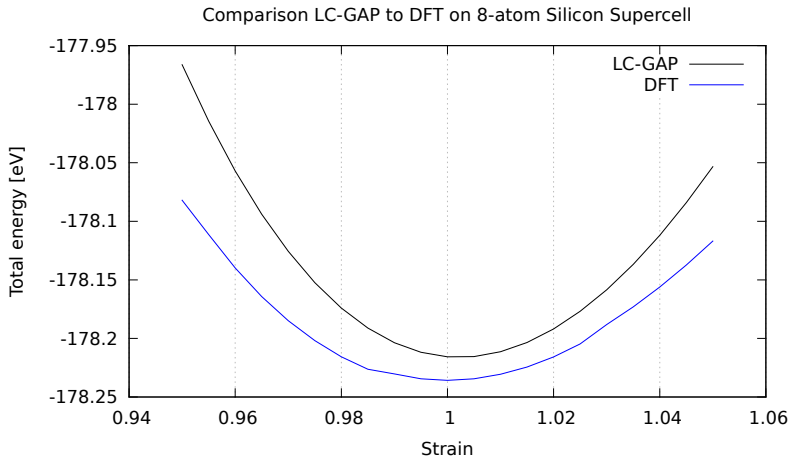


Figure 6.10.: Comparison of the energy curve provided for the perturbed 8-atom supercell using the LC-GAP trained on 1050 Si8ApALV configurations and the DFT calculator. (localised Coulomb matrix UT, $\alpha = 1.0$; $p_A = 0.3\,\mathring{A}$, $p_{LV} = 0.2\,\mathring{A}$; variance Gaussian noise: $2.7 * 10^{-3}$ eV)

and Ghahramani [49]. With this technique, he is able to reduce the training set to 300 atomic neighbourhoods. The incorporation of a sparsification method into our implementation is an important subject for future research.

Last but not least, we compare the energy curve provided by the LC-GAP for the 8-atom silicon supercell to the DFT method in Figure 6.10. Here, the LC-GAP was trained on 1050 Si8ApALV configurations using the localised Coulomb matrix UT with the standard localisation exponent and the hyperparameters for the isotropic Gaussian kernel chosen above as 47.42 ± 0.88 for the amplitude and as 434.61 ± 8.11 for the characteristic length scale. The configurations used for testing were chosen by multiplying the lattice constant by a strain factor between 0.95 and 1.05. We observe that the LC-GAP is able to reproduce the form of the DFT-curve quite well. The absolute deviation in prediction is less than 0.05 eV and most likely due to the basis supercell not being included in the reference data. As noted before, this could be improved in further applications of the LC-GAP.

We now continue the validation of the usefulness of the LC-GAP in molecular dynamics simulations by evaluating its performance when predicting gradient values.

6.2.3. Predicting Gradient Values

As described in Section 3.3.1, differentiating the target function reconstructed via Gaussian process regression as in Equation (3.10) gives a prediction for the gradient values. This is important for molecular dynamics simulations as the negative gradient with respect to the particle coordinates corresponds to the force acting on the particle. We evaluate the accuracy on all three data sets

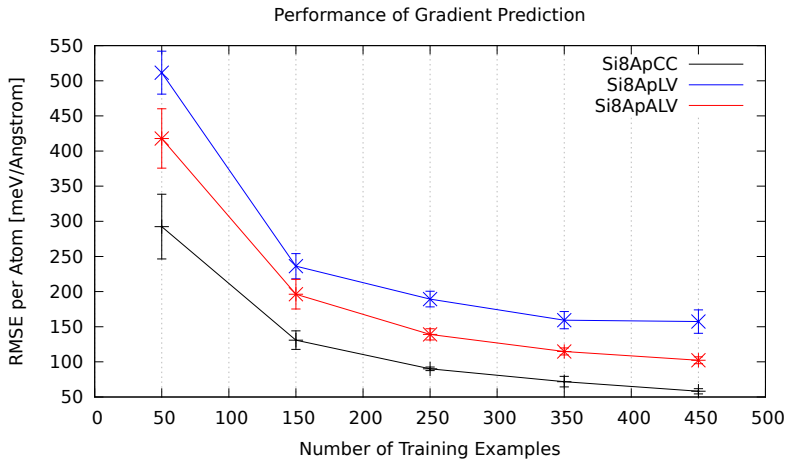


Figure 6.11.: Comparison of the root mean squared errors of gradient prediction on the data sets Si8ApCC, Si8ApLV and Si8ApALV using the localised Coulomb matrix UT, $\alpha = 1.0$ ($p_A = 0.3\,\mathring{A}$, $p_{LV} = 0.5\,\mathring{A}$ (Si8ApLV), $p_{LV} = 0.2\,\mathring{A}$ (Si8ApALV); variance Gaussian noise: $2.7 * 10^{-3}$ eV; No. CV runs: 5)

Si8ApCC, Si8ApLV and Si8ApALV using the localised Coulomb matrix UT with the localisation exponent α set to 1.0. As perturbation values we stay with $0.3\,\mathring{A}$ for the atoms, $0.5\,\mathring{A}$ when perturbing only the lattice vectors and $0.2\,\mathring{A}$ for the lattice vectors when perturbing both. As characteristic length scales we use the values chosen via model selection in the previous subsections.

We have summarised the prediction errors for the gradients in Table 6.30. All in all, we obtain RMS errors per atom that are comparable or even slightly better than those presented by Bartók-Pártay ($<500\,\mathrm{meV}/\mathring{A}$). Learning 450 training configurations leads to values of 58.16 ± 3.66, 157.35 ± 16.67 and $102.22 \pm 4.90\,\mathrm{meV}/\mathring{A}$ per atom on Si8ApCC, Si8ApLV and Si8ApALV, respectively. This is two orders of magnitude worse than our results for the prediction of the energy values.

For better comparison, we plot the RMSEs for the gradient prediction as a function of the training set sizes for the three data sets in Figure 6.11. As additional training set sizes 250 and 350 configurations are chosen, both combined with a test set size of 100 configurations.

As before, we observe that larger training sets, about double in size, are needed on Si8ApLV and Si8ApALV to obtain the same accuracy as on Si8ApCC. Nevertheless, it seems that the accuracy of the gradient prediction saturates for all three data sets rather quickly. For Si8ApCC it saturates at about $50\,\mathrm{meV}/\mathring{A}$ per atom, for Si8ApALV at about $100\,\mathrm{meV}/\mathring{A}$ per atom and for Si8ApLV at about $150\,\mathrm{meV}/\mathring{A}$ per atom. Since we decreased the maximum perturbation value on the lattice vectors when sampling across all dimensions, the prediction of gradient values is better on Si8ApALV than when not simultaneously perturbing the atomic coordinates as well.

It is to be expected that larger training set sizes do not improve the prediction

errors indefinitely, as we include no gradient information in the training data. The incorporation of gradient values into the learning process is an important extension for future research. All the more, this means that the accuracy we obtain in predicting the gradients naively by differentiation of the reconstructed function is astonishing, as it is more than comparable to the results reported by Bartók-Pártay who does include gradient information in the training data.

We conclude from the analysis performed in this section that the localised Coulomb matrix has the potential of being a suitable descriptor for the local atomic environments of crystalline solids as well as of organic molecules. It can be used for both learning across chemical compound space and the interpolation of the PES of a single chemical structure. Concerning the exponent α in the entries of the localised Coulomb matrix, we observe that increasing the localisation effects different chemical structures differently. While it leads to a significant improvement on the biomolecular data sets, there is little to none on the silicon crystals. Hence, we note that the exponent should be included in the hyperparameters of the framework and should be fitted according to the chemical compound in question just as the characteristic length scale. Nevertheless, the standard exponent $\alpha = 1.0$ leads to satisfactory results on both data sets and can be used as a default value.

Data Set	Train. Set Size	Test Set Size	Ampl.	Char. Length Scale	MAE per atom [meV/$\mathring{A}$]	RMSE per atom [meV/$\mathring{A}$]
Si8ApCC	50	10	37.37	459.28	233.82 ± 41.87	292.49 ± 46.00
Si8ApCC	150	50	37.37	459.28	103.48 ± 10.30	130.88 ± 13.18
Si8ApCC	450	100	37.37	459.28	45.40 ± 2.75	**58.16 ± 3.66**
Si8ApLV	50	10	46.33	419.87	400.92 ± 29.11	511.52 ± 30.57
Si8ApLV	150	50	46.33	419.87	182.59 ± 10.55	236.13 ± 17.95
Si8ApLV	450	100	46.33	419.87	115.27 ± 9.59	**157.35 ± 16.67**
Si8ApALV	50	10	47.42	434.61	331.43 ± 31.20	417.89 ± 42.31
Si8ApALV	150	50	47.42	434.61	155.34 ± 17.27	196.27 ± 21.08
Si8ApALV	450	100	47.42	434.61	79.44 ± 4.15	**102.22 ± 4.90**

Table 6.30.: Prediction errors for the gradients using the localised Coulomb matrix UT, $\alpha = 1.0$ ($p_A = 0.3\,\mathring{A}$, $p_{LV} = 0.5\,\mathring{A}$ (Si8ApLV), $p_{LV} = 0.2\,\mathring{A}$ (Si8ApALV); variance Gaussian noise: $2.7 * 10^{-3}$ eV; No. CV runs: 5)

7. Conclusions and Outlook

In this thesis, we presented a kernel-based learning method for the efficient approximation of the Born-Oppenheimer potential energy hypersurface. We combined the Gaussian approximation potentials framework [4] described by Bartók *et al.* in 2010 with a newly introduced local descriptor of atomic environments based on the Coulomb matrix [45] presented by M. Rupp *et al.* in 2012. We now summarise the properties and advantages of this improved approach, which we called *localised Coulomb matrix based Gaussian approximation potentials (LC-GAP)* and discuss possible extensions.

7.1. Conclusion

Building on the Gaussian Approximation Potential (GAP) presented by Bartók *et al.*, we described a new framework for the interpolation of potential energy surfaces by changing the numerical representation of the particle system. Our solution was based on the global molecular descriptor introduced by Rupp *et al.*, which we applied to the local atomic environment of a single particle by weighting all pairwise contributions by their combined distance to the central particle. We called this local version *localised Coulomb matrix*. This descriptor incorporated the required physical invariances with respect to translations and rotations by construction and with respect to atom-indexing by sorting its rows and columns in a suitable way. As the localisation was achieved by measuring all distances only with respect to the central particle, this descriptor encodes no angular information about the neighbours and does not distinguish between environments differing only in local rotations around the central particle.[1]

We demonstrated that the resulting LC-GAP represents the potential energy hypersurface as a linear combination of kernel functions centered at the atomic environments present in the given training data. The forces applied to the system required for molecular dynamics simulations can be obtained by using the gradient of the respective kernel basis functions.

In their recent review [17] of Machine Learning approaches for the potential energy hypersurface interpolation, Handley and Behler state six fundamental requirements for constructing atomistic potentials: Accuracy, Efficiency, Generality, Reactivity, Automation and Costs. We summarise that in the conducted experiments the LC-GAP fulfills all of them.

[1] **Revised Version:** The last sentence of this paragraph was added to address the limitations of the uniqueness of the descriptor.

First, we have shown that the LC-GAP predicts the atomisation energies of organic molecules in equilibrium configuration with an accuracy close to the desired chemical accuracy of 1 kcal/mol. Our result of a mean absolute error of 4.47 ± 0.32 kcal/mol from the DFT data obtained when executing a 5-fold cross validation procedure on only 600 training examples of the QM7 data set is nearly twice as good as the reference value reported by Hansen *et al.* in [20] for more than 10 times as many reference molecules. For alcanes, we were able to reduce the error to 1.56 ± 0.21 kcal/mol.

Additionally, the LC-GAP is transferable as testing a potential trained only on a set of small molecules resulted in a mean absolute error of 7.67 ± 1.12 kcal/mol for larger molecules with up to six atoms more. We stress that this prediction is better than the reference value which was obtained on the larger molecules themselves. This constitutes a strong indication that the localisation ansatz is valid for organic molecules.

Furthermore, predicting the total energies and forces for perturbed silicon supercells using the LC-GAP resulted in an accuracy comparable to that presented by Bartók-Pártay in his initial publication of the GAP [4]. We stress that we obtain this result without including the unperturbed silicon supercell itself or gradient information in the reference data. This demonstrates the versatility of the ansatz.

We derived that the complexity of predicting both the energy and forces of a particle system using the LC-GAP is linear in the number of particles. This means it is efficient enough to enable large scale simulations clearly beyond the realm of DFT methods.

The LC-GAP is general in the sense that it can be successfully applied to both finite and infinitely periodic chemical structures and since no prior assumptions about the bonding structure are made, it is also able to describe arbitrary chemical reactions. It can be applied to learning across chemical compound space (excluding conformational and stereoisomers)[2] as well as to molecular dynamics simulations in the material sciences. Additionally, it could also be used to correct empirical potentials with respect to DFT calculations (see below in the outlook section).

The model parameters that need to be fitted to the reference data can be selected by an automated procedure of nested cross validation and maximisation of the marginal likelihood with a minimum of human effort.

We conclude that our LC-GAP constitutes a promising approach to the interpolation of general potential energy hypersurfaces capable of a prediction accuracy close to DFT calculations but involving only a fraction of their computation time.

[2] **Revised Version:** This restriction was added to address the limitations of the uniqueness of the descriptor.

7.2. Outlook

We now discuss possible enhancements to the LC-GAP which merit further research.

Of course, establishing uniqueness of the representation for any atomic environment, i.e. including angular information between neighbours into the descriptor, would extend the validity of the LC-GAP to biomolecular applications where stereoisomers are common. There are multiple possibilities of achieving this, one of which has been proposed in Section 4.3.2.[3]

One of the most promising extensions to the LC-GAP for applications in the material sciences would be the incorporation of gradient information into the training data, turning the Gaussian process regression into a scheme similar to Hermite interpolation. We have described the mathematical background in Chapter 3. Unfortunately, we were not able to test it due to time constraints. The improvement in prediction accuracy could be expected to be at least one order of magnitude.

This would potentially lead to larger training set sizes necessitating the use of sparsification methods, such as the ansatz proposed by Snelson and Ghahramani in [49].

Aside from these improvements, it would certainly be beneficial to extend the similarity measures employed by the LC-GAP. Instead of focusing only on the Gaussian kernel, a more general choice of the Matérn kernel could improve the prediction. Hansen *et al.* favor the Laplacian kernel for the prediction of atomisation energies in [20] and [19], however, using this kernel for molecular dynamics applications is not advisable as it is not continuously differentiable. Nevertheless, adoption of the Matérn class for possible kernels would include both choices, Gaussian and Laplacian, and could provide a rigorous comparison of their performance among others.

As already described in Section 3.4, the LC-GAP could also be applied to learning energy correction terms between two energy models with different accuracy properties and computational costs. Here the goal would be to achieve the accuracy of e.g. a costly DFT method by learning the difference to any suitable, much faster to evaluate, empirical potential.

This ansatz could also be used to improve the performance of the LC-GAP if the validity of the localisation assumption is doubtful and prediction fails for computationally reasonable cutoff values. This could be the case when long-range interactions are present, as then the size of the atomic neighbourhood has to be very large in order to capture all relevant information. We would then account for the long-range interactions explicitly using a long-range Coulomb potential and learn only the short-range part of the total energy via the LC-GAP, thereby reinstating the validity of the localisation assumption.

[3] **Revised Version:** This paragraph was added to address the possibility of extending the uniqueness of the localised Coulomb matrix to any atomic environment.

A similar, yet more fundamental extension could be done concerning the underlying decomposition ansatz for the energy. Instead of using the atomistic decomposition, we could consider the many-body cluster expansion,

$$E_{\text{total}} = \sum_i E^{(i)}_{\text{atomic}} + \sum_{i<j} E^{(ij)}_{\text{pair}} + \sum_{i<j<k} E^{(ijk)}_{\text{triple}} + ... + E_{\text{n-tuple}}, \tag{7.1}$$

which can also be interpreted as an ANOVA-like decomposition or High-dimensional Model Representation (HDMR) [21]. We could then use the difference learning ansatz to impose a hierarchical structure of energy approximation models. Based on the prediction of the atomistic energy contributions, one could learn the pairwise contributions as a correction term to the total energy in a second step. The procedure could then be iterated such that the contribution of a j-tuple is learned as the difference between the total energy and all interactions stemming from less than j atoms. This way the prediction could be improved step-wise until a predefined tolerance is achieved.

We summarise that the LC-GAP in its current state is able to generate reliable potentials for large scale simulations in both biochemistry and the material sciences. Nonetheless, it is flexible enough to handle future challenges through further extensions, since the above enhancements can be incorporated into the LC-GAP in a modular fashion.

List of Figures

List of Tables

Listings

A. Basic Stochastic Concepts

In this chapter we state some basic concepts from probability theory. For details refer to [25].

Definition 4 (Probability Space)**.**
A probability space is a triple $(\Omega, \mathcal{F}, P)$ *consisting of an non-empty set* Ω*, a* σ*-algebra* $\mathcal{F} \subseteq 2^{\Omega}$ *and a probability measure* $P : \mathcal{F} \to [0, 1]$*. Here,* Ω *is called the sample space and the* σ*-algebra* $\mathcal{F}$ *constitues a set of subsets of* Ω*, called measurable sets, such that*

- $\Omega \in \mathcal{F}$*,*
- *if* $A \in \mathcal{F}$*, then* $(\Omega \setminus A) \in \mathcal{F}$*,*
- *if* $A_i \in \mathcal{F}$ *for* $i \in I$ *with* I *a countable index set, then* $(\cup_{i \in I} A_i) \in \mathcal{F}$*.*

Furthermore, a probability measure P *is a function assigning each measurable set* $A \in \mathcal{F}$ *a non-negative number* $P(A)$ *such that* $P(\Omega) = 1$ *and* P *is countably additive, i.e. if* $\{A_i\} \subseteq \mathcal{F}$ *is a countable collection of disjoint sets, then*

$$P(\cup A_i) = \sum P(A_i).$$

Definition 5 (Random Variable)**.**
A random variable $Y : \Omega \to S$ *on a given probability space* $(\Omega, \mathcal{F}, P)$ *is defined as a measurable function from the sample space* Ω *to another measurable space* $(S, \mathcal{S})$ *called state space. Here, the term measurable function means that the preimages of measurable sets are measurable, i.e.,* $Y^{(-1)}(B) \in \mathcal{F}$ *for all* $B \in \mathcal{S}$*.*

Its probability distribution is defined as the measure P_Y *on* $(S, \mathcal{S})$ *given by*

$$P_Y(A) := P(\{\omega \in \Omega : Y(\omega) \in A\}), \text{ for } A \in \mathcal{S}. \tag{A.1}$$

Definition 6 (Expectation Value)**.**
The expectation value or mean of a random variable $Y : \Omega \to S$ *on a probability space* $(\Omega, \mathcal{F}, P)$ *is defined as the Lebesgue integral*

$$\mathbb{E}[Y] = \int_{\Omega} Y(\omega) dP(\omega). \tag{A.2}$$

If the probability distribution of Y *admits a probability density* ρ *with respect to the Lebesgue measure, i.e.,*

$$P(A) = \int_A \rho(s) ds \tag{A.3}$$

then the expected value can be written as

$$\mathbb{E}[Y] = \int_S x\rho(s)ds. \tag{A.4}$$

B. Gaussian Identities for the Multivariate Normal Distribution

B.1. The Multivariate Normal Distribution

We start by explicitely stating the density function of the multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$ with mean vector μ and covariance matrix Σ. For an n-dimensional vector x it has the form

$$p(x) = \frac{1}{(2\pi)^{n/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right) \tag{B.1}$$

B.2. Conditional Gaussian Distributions

The identities derived here for the multivariate normal distribution follow the exposition from the book [8] by Bishop, Chapter 2.3.

If two vectors of random variables are jointly Gaussian, conditioning one vector on the other will again result in a normal distribution. In order to derive its mean and its covariance matrix, consider the vector x composed of two disjoint subvectors x_a and x_b as

$$x = \begin{pmatrix} x_a \\ x_b \end{pmatrix}. \tag{B.2}$$

If x is normally distributed with mean vector μ and covariance matrix Σ, we partion both accordingly as

$$\mu = \begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}. \tag{B.3}$$

For simplicity we denote the inverse of the covariance matrix by Θ and partition it as

$$\Theta = \begin{pmatrix} \Theta_{aa} & \Theta_{ab} \\ \Theta_{ba} & \Theta_{bb} \end{pmatrix}. \tag{B.4}$$

Using Schur's complement for Σ, it holds

$$\begin{aligned}
\Theta_{aa} &= (\Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba})^{-1}, \\
\Theta_{ab} &= -(\Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba})^{-1}\Sigma_{ab}\Sigma_{bb}^{-1}, \\
\Theta_{ba} &= -\Sigma_{bb}^{-1}\Sigma_{ba}(\Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba})^{-1}, \\
\Theta_{bb} &= \Sigma_{bb}^{-1} + \Sigma_{bb}^{-1}\Sigma_{ba}(\Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba})^{-1}\Sigma_{ab}\Sigma_{bb}^{-1}.
\end{aligned} \tag{B.5}$$

The conditional distribution of x_a given x_b is defined as

$$p(x_a|x_b) = \frac{p(x_a, x_b)}{p(x_b)}. \tag{B.6}$$

This means it can be determined by considering $p(x_a, x_b)$ only as a function of x_a with x_b fixed and then normalising it in a suitable way. To this end, we write the functional dependence in x of the normal distribution as

$$(x-\mu)^T\Sigma^{-1}(x-\mu) = x_a^T\Theta_{aa}x_a + x_a(\Theta_{aa}\mu_a - \Theta_{ab}(x_b - \mu_b)) + const, \tag{B.7}$$

where we have grouped the terms according to their power in x_a. Comparison of the coefficients leads to the equations

$$\begin{aligned}
\Sigma_{a|b} &= \Theta_{aa}^{-1}, \\
\Sigma_{a|b}^{-1}\mu_{a|b} &= \Theta_{aa}\mu_a + \Theta_{ab}(x_b - \mu_b),
\end{aligned} \tag{B.8}$$

From those we obtain as mean and covariance matrix of the conditional distributions

$$\begin{aligned}
\mu_{a|b} &= \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(x_b - \mu_b), \\
\Sigma_{a|b} &= \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}.
\end{aligned} \tag{B.9}$$

C. Calculation Details

C.1. Derivatives of the Localised Kernel

When incorporating gradient information into the Gaussian Process regression, it is necessary to compute first and second order derivatives of the kernel function with respect to the cartesian coordinates of the particles. As the kernels are defined as functions of the local atomic environments, this implies the need for the application of the chain rule. Consequently, one computes for $m, n = 1, ..., K$, $j = 1, ..., N$ and $l = 1, ..., P_j$,

$$\begin{aligned}\frac{\partial}{\partial x_j^l} \mathcal{k}(q_m(X), q_n(X)) &= \frac{\partial \mathcal{k}(q_m(X), q_n(X))}{\partial q_m} \cdot \frac{\partial q_m}{\partial x_j^l} + \frac{\partial \mathcal{k}(q_m(X), q_n(X))}{\partial q_n} \cdot \frac{\partial q_n}{\partial x_j^l} \\ &= \frac{\partial (\mathbf{C}_K)_{mn}}{\partial \mathbf{q}} \cdot \frac{\partial \mathbf{q}}{\partial x_j^l},\end{aligned} \tag{C.1}$$

and

$$\begin{aligned}\frac{\partial}{\partial x_i^k}\frac{\partial}{\partial x_j^l} \mathcal{k}(q_m(X), q_n(X)) &= \frac{\partial}{\partial x_i^k}\Big(\frac{\partial (\mathbf{C}_K)_{mn}}{\partial \mathbf{q}} \cdot \frac{\partial \mathbf{q}}{\partial x_j^l}\Big) \\ &= \frac{\partial \mathbf{q}}{\partial x_i^k} \cdot \frac{\partial^2 (\mathbf{C}_K)_{mn}}{\partial \mathbf{q} \partial \mathbf{q}} \cdot \frac{\partial \mathbf{q}}{\partial x_j^l} + \frac{\partial (\mathbf{C}_K)_{mn}}{\partial \mathbf{q}} \cdot \frac{\partial^2 \mathbf{q}}{\partial x_i^k \partial x_j^l}.\end{aligned} \tag{C.2}$$

This means that when defining a local environment representation $\mathbf{q}$, one also needs to specify the derivatives $\frac{\partial \mathbf{q}}{\partial x_j^l}$ and $\frac{\partial^2 \mathbf{q}}{\partial x_i^k \partial x_j^l}$.

Similarly, calculating the derivatives of the kernel applied to test and training systems both with respect to the coordinates of a training and of a test system, one obtains,

$$\begin{aligned}\frac{\partial \mathcal{k}(q_i, q_p^\star)}{\partial x_n^k} &= \frac{\partial (\mathbf{c}_{(p)}^\star)_i}{\partial \mathbf{q}} \cdot \frac{\partial \mathbf{q}}{\partial x_n^k}, \\ \frac{\partial \mathcal{k}(q_i, q_p^\star)}{\partial x_\star^k} &= \frac{\partial (\mathbf{c}_{(p)}^\star)_i}{\partial \mathbf{q}^\star} \cdot \frac{\partial \mathbf{q}^\star}{\partial x_\star^k},\end{aligned} \tag{C.3}$$

and

$$\frac{\partial^2 \mathcal{k}(q_p^\star, q_l)}{\partial x_\star^i \partial x_n^k} = \Big(\frac{\partial \mathbf{q}^\star}{\partial x_\star^i}\Big)^T \cdot \frac{\partial^2 (\mathbf{c}_{(p)}^\star)_l}{\partial \mathbf{q} \partial \mathbf{q}^\star} \cdot \frac{\partial \mathbf{q}}{\partial x_n^k}. \tag{C.4}$$

C.2. Entries of the Extended Covariance Matrices

C.2.1. Covariance between Training Data, Function Values and Gradients

The single entries describing the covariance between energy and gradient values can be computed as follows,

$$\begin{aligned}\mathrm{Cov}(E_{\mathrm{total}}(\mathbf{x}_i), g_j^l) &= \frac{\partial}{\partial x_j^l}\mathrm{Cov}(E_{\mathrm{total}}(\mathbf{x}_i), E_{\mathrm{total}}(\mathbf{x}_j)) \\ &= \frac{\partial}{\partial x_j^k}(\mathbf{L}^T\mathbf{C}_K\mathbf{L})_{ij} \\ &= \sum_{m,n=1}^{K} L_{mi}\frac{\partial}{\partial x_j^l}\mathcal{k}(q_m, q_n)L_{nj} \\ &= \sum_{m,n=1}^{K} L_{mi}\Big(\frac{\partial(\mathbf{C}_K)_{mn}}{\partial\mathbf{q}}\cdot\frac{\partial\mathbf{q}}{\partial x_j^l}\Big)L_{nj},\end{aligned} \tag{C.5}$$

where the derivative of the kernel was given in Appendix C.1. Collecting the derivatives for a given cartesian coordinate x_j^l in a matrix $\mathbf{D}^{j,l}$,

$$\mathbf{D}^{j,l} := \Big(\frac{\partial(\mathbf{C}_K)_{lm}}{\partial\mathbf{q}}\cdot\frac{\partial\mathbf{q}}{\partial x_j^l}\Big)_{l,m=1}^{K}, \tag{C.6}$$

one obtains as an expression for the whole block

$$\begin{aligned}\mathbf{C}_{f,g_j} &= \mathrm{Cov}(\mathbf{E}_{\mathrm{total}}, \mathbf{g}_j) \\ &= \begin{pmatrix} \sum_{m,n=1}^{K} L_{m1}(\mathbf{D}^{j,1})_{mn}L_{nj} & \dots & \sum_{m,n=1}^{K} L_{m1}(\mathbf{D}^{j,P_j})_{mn}L_{nj} \\ \vdots & & \vdots \\ \sum_{l,m=1}^{K} L_{mN}(\mathbf{D}^{j,1})_{mn}L_{nj} & \dots & \sum_{m,n=1}^{K} L_{mN}(\mathbf{D}^{j,P_j})_{mn}L_{nj} \end{pmatrix} \\ &= \mathbf{L}^T\Big(\mathbf{D}^{j1}\mathbf{L}_{:,j}, \dots, \mathbf{D}^{jP_j}\mathbf{L}_{:,j}\Big).\end{aligned} \tag{C.7}$$

Analogously one obtains,

$$\mathbf{C}_{g_i,f} = \mathrm{Cov}(\mathbf{g}_i, \mathbf{E}_{\mathrm{total}}) = \begin{pmatrix} \mathbf{L}_{i,:}^T\mathbf{D}^{i1} \\ \vdots \\ \mathbf{L}_{i,:}^T\mathbf{D}^{iP_i} \end{pmatrix}\mathbf{L}. \tag{C.8}$$

This can be written in a more compact way by defining a block matrix $\mathbf{D}$ with the matrices $\mathbf{D}^{i1}, \dots, \mathbf{D}^{iP_i}$ as row i and introducing a shorthand notation for the multiplication of its rows with a column vector or row vector, i. e.,

$$\mathbf{D}^{j,:}\mathbf{L}_{:j} := \Big(\mathbf{D}^{j1}\mathbf{L}_{:,j}, \dots, \mathbf{D}^{jP_j}\mathbf{L}_{:,j}\Big), \tag{C.9}$$

and

$$\mathbf{L}_{i,:}^T \mathbf{D}^{i,:} := \begin{pmatrix} \mathbf{L}_{i,:}^T \mathbf{D}^{i1} \\ \vdots \\ \mathbf{L}_{i,:}^T \mathbf{D}^{iP_i} \end{pmatrix}. \tag{C.10}$$

The expression for the covariance between gradient values is more complicated as it involves second derivatives.

$$\begin{aligned} \mathrm{Cov}(g_i^k, g_j^l) &= \mathrm{Cov}\Big(\frac{\partial E_{\text{total}}(\mathbf{x}_i)}{\partial x_i^k}, \frac{\partial E_{\text{total}}(\mathbf{x}_j)}{\partial x_j^l}\Big) \\ &= \frac{\partial}{\partial x_i^k}\frac{\partial}{\partial x_j^l}(\mathbf{L}^T \mathbf{C}_K \mathbf{L})_{ij} \\ &= \sum_{m,n=1}^{K} L_{mi} \frac{\partial}{\partial x_i^k}\frac{\partial}{\partial x_j^l} \mathcal{k}(q_m, q_n) L_{nj} \\ &= \sum_{m,n=1}^{K} L_{mi} \Big(\frac{\partial \mathbf{q}}{\partial x_i^k} \cdot \frac{\partial^2 (\mathbf{C}_K)_{mn}}{\partial \mathbf{q} \partial \mathbf{q}} \cdot \frac{\partial \mathbf{q}}{\partial x_j^l} + \frac{\partial (\mathbf{C}_K)_{mn}}{\partial \mathbf{q}} \cdot \frac{\partial^2 \mathbf{q}}{\partial x_i^k \partial x_j^l} \Big) L_{nj} \end{aligned} \tag{C.11}$$

Defining for every index pair (i, j) a block matrix $\mathbf{H}_{(i,j)}$ comprised of matrices

$$\mathbf{H}_{(i,j)}^{kl} := \Big(\frac{\partial \mathbf{q}}{\partial x_i^k} \cdot \frac{\partial^2 (\mathbf{C}_K)_{mn}}{\partial \mathbf{q} \partial \mathbf{q}} \cdot \frac{\partial \mathbf{q}}{\partial x_j^l} + \frac{\partial (\mathbf{C}_K)_{mn}}{\partial \mathbf{q}} \cdot \frac{\partial^2 \mathbf{q}}{\partial x_i^k \partial x_j^l} \Big)_{m,n=1}^{K}, \tag{C.12}$$

and extending the shorthand notation, it holds

$$\mathbf{C}_{g_i, g_j} = \mathrm{Cov}(\mathbf{g}_i, \mathbf{g}_j) = \begin{pmatrix} \mathbf{L}_{i,:}^T \mathbf{H}_{(i,j)}^{1,:} \mathbf{L}_{:,j} \\ \vdots \\ \mathbf{L}_{i,:}^T \mathbf{H}_{(i,j)}^{P_i,:} \mathbf{L}_{:,j} \end{pmatrix} =: \mathbf{L}_{i,:}^T \mathbf{H}_{(i,j)} \mathbf{L}_{:,j}. \tag{C.13}$$

Summarizing, the extended covariance matrix can be written as

$$\mathbf{C}_{\text{ext}} = \begin{pmatrix} L^T \mathbf{C}_K L & \mathbf{L}^T \mathbf{D}^{1,:} \mathbf{L}_{:1} & \dots & \mathbf{L}^T \mathbf{D}^{N,:} \mathbf{L}_{:N} \\ \mathbf{L}_{1,:}^T \mathbf{D}^{1,:} \mathbf{L} & \mathbf{L}_{1,:}^T \mathbf{H}_{(1,1)} \mathbf{L}_{:,1} & \dots & \mathbf{L}_{1,:}^T \mathbf{H}_{(1,N)} \mathbf{L}_{:,N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{L}_{N,:}^T \mathbf{D}^{N,:} \mathbf{L} & \mathbf{L}_{N,:}^T \mathbf{H}_{(N,1)} \mathbf{L}_{:,1} & \dots & \mathbf{L}_{N,:}^T \mathbf{H}_{(N,N)} \mathbf{L}_{:,N} \end{pmatrix}. \tag{C.14}$$

C.2.2. Covariance between Training and Test Data

As with the localised Gaussian process regression applied only to function values, one has

$$\mathbf{C}^{\star}_{f^{\star},f} = \sum_{p=1}^{P^{\star}} (\mathbf{c}^{\star}_{(p)})^{T} \mathbf{L}. \tag{C.15}$$

The same technique as used in Subsection 3.2.1 can be applied to compute

$$\begin{aligned} \mathrm{Cov}(E^{\star}_{\mathrm{total}}, \frac{\partial E_{\mathrm{total}}(\mathbf{x}_n)}{\partial x^k_n}) &= \frac{\partial}{\partial x^k_n} \mathrm{Cov}(\sum_{p=1}^{P^{\star}} f(q^{\star}_p), \sum_{i=1}^{K} f_i L_{in}) \\ &= \sum_{p=1}^{P^{\star}} \sum_{i=1}^{K} \frac{\partial \mathcal{k}(q_i, q^{\star}_p)}{\partial x^k_n} L_{in}. \end{aligned} \tag{C.16}$$

Introducing for a given particle p and training system n the matrices $\mathbf{d}^n_{(p)}$ comprised of the derivatives of the local kernels with respect to the training system coordinates,

$$(\mathbf{d}^n_{(p)})_{ik} = \frac{\partial \mathcal{k}(q_i, q^{\star}_p)}{\partial \mathbf{q}} \cdot \frac{\partial \mathbf{q}}{\partial x^k_n}, \tag{C.17}$$

it holds

$$\mathbf{C}^{\star}_{f^{\star},g_n} = \sum_{p=1}^{P^{\star}} \mathbf{L}^{T}_{n,:} \mathbf{d}^n_{(p)}. \tag{C.18}$$

Analogously, one can define matrices $\mathbf{d}^{\star}_{(p)}$ comprised of the derivatives of the local kernels with respect to the test system coordinates,

$$(\mathbf{d}^{\star}_{(p)})_{ik} = \frac{\partial \mathcal{k}(q_i, q^{\star}_p)}{\partial \mathbf{q}^{\star}} \cdot \frac{\partial \mathbf{q}^{\star}}{\partial x^k_{\star}}, \tag{C.19}$$

to obtain

$$\mathbf{C}^{\star}_{g^{\star},f} = \sum_{p=1}^{P^{\star}} (\mathbf{d}^{\star}_{(p)})^{T} \mathbf{L} \tag{C.20}$$

as an expression for the covariance between test gradients and learnt energy values.

Lastly, collecting the second order derivatives of the local kernels with respect to the test system coordinates and a given training system coordinate x^k_n into matrices $\mathbf{h}^{nk}_{(p)}$ for a given particle p,

$$(\mathbf{h}^{nk}_{(p)})_{il} = \Big(\frac{\partial \mathbf{q}^{\star}}{\partial x^i_{\star}}\Big)^{T} \cdot \frac{\partial^2 \mathcal{k}(q^{\star}_p, q_l)}{\partial \mathbf{q} \partial \mathbf{q}^{\star}} \cdot \frac{\partial \mathbf{q}}{\partial x^k_n}, \tag{C.21}$$

and building a block matrix $\mathbf{h}_{(p)}$ out of these submatrices, one can write the covariance between gradients as

$$\mathbf{C}^{\star}_{g^{\star},g_n} = \sum_{p=1}^{P^{\star}} \mathbf{h}^{n,:}_{(p)} \mathbf{L}_{:,n}. \tag{C.22}$$

To summarize, the extended covariance matrix between training and test data including gradient information can be written as

$$\mathbf{c}_{\text{ext}} = \sum_{p=1}^{P^{\star}} \begin{pmatrix} (\mathbf{c}^{\star}_{(p)})^T \mathbf{L} & \mathbf{L}^T_{1,:} \mathbf{d}^1_{(p)} & \dots & \mathbf{L}^T_{N,:} \mathbf{d}^N_{(p)} \\ (\mathbf{d}^{\star}_{(p)})^T \mathbf{L} & \mathbf{h}^{1,:}_{(p)} \mathbf{L}_{:,1} & \dots & \mathbf{h}^{N,:}_{(p)} \mathbf{L}_{:,N} \end{pmatrix}. \tag{C.23}$$

Bibliography

[1] Bartók-Pártay, A. *The Gaussian Approximation Potential: an interatomic potential derived from first principles Quantum Mechanics.* Springer Theses. Springer, Berlin Heidelberg, 2010.

[2] Bartók, A. P., Gillan, M. J., Manby, F. R., and Csányi, G. Machine-learning approach for one-and two-body corrections to density functional theory: Applications to molecular and condensed water. *Physical Review B 88*, 5 (2013), 054104.

[3] Bartók, A. P., Kondor, R., and Csányi, G. On representing chemical environments. *Physical Review B 87*, 18 (2013), 184115.

[4] Bartók, A. P., Payne, M. C., Kondor, R., and Csányi, G. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Physical review letters 104*, 13 (2010), 136403.

[5] Behler, J. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *The Journal of chemical physics 134*, 7 (2011), 074106.

[6] Behler, J., and Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett. 98* (2007), 146401.

[7] Bettens, R. P., and Collins, M. A. Learning to interpolate molecular potential energy surfaces with confidence: A bayesian approach. *The Journal of chemical physics 111*, 3 (1999), 816–826.

[8] Bishop, C. M., et al. *Pattern recognition and machine learning.* Springer, New York, 2006.

[9] Blum, L. C., and Reymond, J.-L. 970 million druglike small molecules for virtual screening in the chemical universe database gdb-13. *Journal of the American Chemical Society 131*, 25 (2009), 8732–8733.

[10] Brandbyge, M., Mozos, J.-L., Ordejón, P., Taylor, J., and Stokbro, K. Density-functional method for nonequilibrium electron transport. *Phys. Rev. B 65* (2002), 165401.

[11] Cawley, G. C., and Talbot, N. L. On over-fitting in model selection and subsequent selection bias in performance evaluation. *The Journal of Machine Learning Research 11* (2010), 2079–2107.

[12] Curtarolo, S., Hart, G. L., Nardelli, M. B., Mingo, N., Sanvito, S., and Levy, O. The high-throughput highway to computational materials design. *Nature materials 12*, 3 (2013), 191–201.

[13] Curtarolo, S., Morgan, D., Persson, K., Rodgers, J., and Ceder, G. Predicting crystal structures with data mining of quantum calculations. *Physical review letters 91*, 13 (2003), 135503.

[14] Engl, H. W., Hanke, M., and Neubauer, A. *Regularization of inverse problems*, vol. 375 of *Mathematics and Its Applications*. Kluwer Academic Publishers, Dordrecht, 2000.

[15] Fischer, C. C., Tibbetts, K. J., Morgan, D., and Ceder, G. Predicting crystal structure by merging data mining with quantum mechanics. *Nature materials 5*, 8 (2006), 641–646.

[16] Griebel, M., Knapek, S., and Zumbusch, G. *Numerical simulation in molecular dynamics: numerics, algorithms, parallelization, applications*, vol. 5 of *Texts in Computational Science and Engineering*. Springer Science & Business Media, Heidelberg, 2007.

[17] Handley, C. M., and Behler, J. Next generation interatomic potentials for condensed systems. *The European Physical Journal B 87*, 7 (2014), 1–16.

[18] Handley, C. M., and Popelier, P. L. Potential energy surfaces fitted by artificial neural networks. *The Journal of Physical Chemistry A 114*, 10 (2010), 3371–3383.

[19] Hansen, K., Biegler, F., von Lilienfeld, O. A., Müller, K.-R., and Tkatchenko, A. Interaction potentials in molecules and non-local information in chemical space. *Phys. Rev. Lett* (Submitted in March 2014).

[20] Hansen, K., Montavon, G., Biegler, F., Fazli, S., Rupp, M., Scheffler, M., von Lilienfeld, O. A., Tkatchenko, A., and Müller, K.-R. Assessment and validation of machine learning methods for predicting molecular atomization energies. *Journal of Chemical Theory and Computation 9*, 8 (2013), 3404–3419.

[21] Hayes, M., Li, B., and Rabitz, H. Estimation of molecular properties by high-dimensional model representation. *Journal of Physical Chemistry 110* (2006), 264–272.

[22] Ho, T.-S., and Rabitz, H. A general method for constructing multidimensional molecular potential energy surfaces from abinitio calculations. *The Journal of chemical physics 104*, 7 (1996), 2584–2597.

[23] Johnson, S. G. Nlopt nonlinear-optimisation package.

[24] Kirsch, A. *An introduction to the mathematical theory of inverse problems*, vol. 120 of *Applied mathematical sciences*. Springer-Verlag, New York, 1996.

[25] Klenke, A. *Wahrscheinlichkeitstheorie*. Springer, Berlin, 2006.

[26] Kohavi, R., et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the International Joint Conference on Artificial Intelligence* (1995), vol. 14, pp. 1137–1145.

[27] Lewars, E. *Computational chemistry: introduction to the theory and applications of molecular and quantum mechanics*. Springer Science & Business Media, Heidelberg, 2010.

[28] Lorenz, S., Scheffler, M., and Gross, A. Descriptions of surface chemical reactions using a neural network representation of the potential-energy surface. *Phys. Rev. B 73* (2006), 115431.

[29] Louis, A. K. *Inverse und schlecht gestellte Probleme*. Teubner, Stuttgart, 1989.

[30] Matérn, B., et al. Spatial variation. stochastic models and their application to some problems in forest surveys and other sampling investigations. *Meddelanden fran statens Skogsforskningsinstitut 49*, 5 (1960).

[31] Mitchell, T. *Machine learning (international edition)*. Mcgraw-Hill, Boston, MA, 1997.

[32] Montavon, G., Hansen, K., Fazli, S., Rupp, M., Biegler, F., Ziehe, A., Tkatchenko, A., Lilienfeld, A. V., and Müller, K.-R. Learning invariant representations of molecules for atomization energy prediction. In *Advances in Neural Information Processing Systems* (2012), pp. 440–448.

[33] Montavon, G., Rupp, M., Gobre, V., Vazquez-Mayagoitia, A., Hansen, K., Tkatchenko, A., Müller, K.-R., and von Lilienfeld, O. A. Machine learning of molecular electronic properties in chemical compound space. *New Journal of Physics 15*, 9 (2013), 095003.

[34] Morawietz, T., and Behler, J. A density-functional theory-based neural network potential for water clusters including van der waals corrections. *The Journal of Physical Chemistry A 117*, 32 (2013), 7356–7366.

[35] Nobelprize.org. The nobel prize in chemistry 2013, 23 April 2015.

[36] Parr, R. G., and Yang, W. *Density-functional theory of atoms and molecules*. Oxford University Press, New York, 1989.

[37] Plimpton, S. J., and Thompson, A. P. Computational aspects of many-body potentials. *MRS bulletin 37*, 05 (2012), 513–521.

[38] Powell, M. J. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in optimization and numerical analysis*. Kluwer Academic, Dordrecht, 1994, pp. 51–67.

[39] Pozun, Z. D., Hansen, K., Sheppard, D., Rupp, M., Müller, K.-R., and Henkelman, G. Optimizing transition states via kernel-based machine learning. *The Journal of chemical physics 136*, 17 (2012), 174101.

[40] quantum-machine.org. QM7 dataset, 26 April 2015.

[41] QuantumWise A/S. Atomistic toolkit (atk) version 2014.2.

[42] Rasmussen, C. E. *Gaussian processes for machine learning*. MIT press, London, 2006.

[43] Robert, C., and Casella, G. *Monte Carlo statistical methods*. Springer Science & Business Media, Heidelberg, 2013.

[44] Rodriguez, J. D., Perez, A., and Lozano, J. A. Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*, 3 (2010), 569–575.

[45] Rupp, M., Tkatchenko, A., Müller, K.-R., and von Lilienfeld, O. A. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical review letters 108*, 5 (2012), 058301.

[46] Scherz, U. *Quantenmechanik: eine Einführung mit Anwendungen auf Atome, Moleküle und Festkörper*. B. G. Teubner, Stuttgart, Leipzig, 1999.

[47] Schölkopf, B., and Smola, A. J. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, London, 2002.

[48] Schütt, K., Glawe, H., Brockherde, F., Sanna, A., Müller, K., and Gross, E. How to represent crystal structures for machine learning: towards fast prediction of electronic properties. *Physical Review B 89*, 20 (2014), 205118.

[49] Snelson, E., and Ghahramani, Z. Sparse gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems* (Cambridge, MA, 2006), vol. 18, MIT Press, pp. 1257–1264.

[50] Snyder, J. C., Rupp, M., Hansen, K., Müller, K.-R., and Burke, K. Finding density functionals with machine learning. *Physical review letters 108*, 25 (2012), 253002.

[51] Soler, J. M., Artacho, E., Gale, J. D., García, A., Junquera, J., Ordejón, P., and Sánchez-Portal, D. The siesta method for ab initio order- n materials simulation. *Journal of Physics: Condensed Matter 14*, 11 (2002), 2745.

[52] Stone, M. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)* (1974), 111–147.

[53] Szlachta, W. J., Bartók, A. P., and Csányi, G. Accuracy and transferability of gaussian approximation potential models for tungsten. *Physical Review B 90*, 10 (2014), 104108.

[54] Thompson, A., Swiler, L., Trott, C., Foiles, S., and Tucker, G. Snap: Automated generation of quantum-accurate interatomic potentials. *Journal of Computational Physics* (2014).

[55] Todeschini, R., and Consonni, V. *Handbook of molecular descriptors*, vol. 11. John Wiley & Sons, Weinheim, 2008.

[56] von Lilienfeld, O. A., Rupp, M., and Knoll, A. Fingerprint representation of molecules: Fourier series of radial distribution functions as descriptor for machine learning across chemical compound space. *arXiv preprint arXiv:1307.2918* (2013).

[57] Willmott, C. J., and Matsuura, K. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research 30*, 1 (2005), 79.

[58] Wright, S. J., and Nocedal, J. *Numerical optimization*. Springer, New York, 1999.